

# 廈門大學



## 信息学院软件工程系

### 《物联网技术导论》大作业报告

题    目         基于 UDP 的物联网环境监测        

专    业                         软件工程                        

年    级                         2021 级                        

姓    名                         顾畅、陈新                        

2024 年 6 月 14 日

## 目录

1	项目背景与意义	- 4 -
1.1	背景	- 4 -
1.2	意义	- 4 -
2	任务概述	- 5 -
2.1	目标	- 5 -
2.2	运行环境	- 5 -
3	数据库设计	- 6 -
3.1	数据表设计	- 6 -
3.2	数据字典	- 6 -
3.3	数据库介绍	- 7 -
4	系统架构设计	- 8 -
4.1	系统总体架构	- 8 -
4.2	技术架构	- 9 -
5	接口定义	- 10 -
6	用户界面设计	- 11 -
6.1	主界面设计	- 11 -
6.2	图表缩放	- 12 -
7	模块设计	- 13 -
7.1	UDPSend	- 13 -
7.1.1	数据生成	- 13 -
7.1.2	UDP 数据传输	- 13 -
7.1.3	主循环逻辑	- 13 -
7.1.4	异常处理	- 14 -
7.2	UDPReceive	- 14 -
7.2.1	UDP 数据接收	- 14 -
7.2.2	数据转发至 Web 服务器	- 14 -
7.2.3	技术实现	- 14 -
7.3	数据可视化	- 15 -

8	备注 .....	- 16 -
8.1	实验人员.....	- 16 -
8.2	GIT 仓库.....	- 16 -

# 1 项目背景与意义

## 1.1 背景

随着城市化进程的不断推进和工业化的发展，环境污染问题日益严重，环境质量的监测和管理成为亟待解决的重要课题。传统的环境监测手段存在数据采集不及时、监测覆盖面有限、人工成本高等问题。物联网（Internet of Things, IoT）技术的发展为环境监测提供了新的解决方案。通过物联网技术，可以实现对环境数据的实时采集、传输和处理，极大地提高环境监测的效率和精度。

本项目旨在模拟利用传感器网络和数据分析技术，对空气质量、水质、噪声等环境参数进行全面监测。通过部署在不同地点的传感器节点，实时收集环境数据，并通过无线网络将数据传输到集中管理平台。数据经过处理和分析，可以为环境管理部门提供及时、准确的决策支持。

## 1.2 意义

1. 实时监测与预警：物联网技术可以实现对环境参数的实时监测，当某些指标超过预设阈值时，可以及时发出预警信号，帮助相关部门迅速采取应对措施，避免环境问题的进一步恶化。
2. 提高监测精度和覆盖范围：物联网环境数据监测项目通过大量分布式传感器节点，实现对广泛区域的环境监测，提高了数据的精度和覆盖范围，弥补了传统监测手段的不足。
3. 数据驱动决策：通过对收集到的大量环境数据进行分析，可以发现环境变化的规律和趋势，为环境管理部门提供科学的决策依据，支持环境保护和治理工作的开展。
4. 降低监测成本：物联网技术的应用可以大幅降低环境监测的人工成本和设备维护成本。自动化的数据采集和传输过程减少了人力投入，提高了监测工作的效率。

## 2 任务概述

### 2.1 目标

本项目旨在使用Java构建一个基于UDP的物联网环境监测项目，可以在所有JVM上运行，提升项目跨平台能力，主要实现以下四个模块：

1. 发送端模拟温度、湿度、照度、二氧化碳浓度、PM2.5浓度、PM10浓度、甲醛含量、噪音等级等环境数据，使用UDP协议将环境数据发送到接收端10000端口。
2. 接收端监听10000端口，接收来自发送端的UDP数据包，解析出环境数据，并调用后端API将数据写入数据库。
3. 后端采用JavaWeb实现，可以支持数据库数据存取等操作。
4. 前段调用后端提供的API，并将返回数据以图表形式可视化展现。

### 2.2 运行环境

开发环境：JDK 17；

操作系统：Mac OS 14.5；

数据库：MySQL 8.0。

### 3 数据库设计

#### 3.1 数据表设计

environment_data		
PK	id	INTEGER
	record_time	VARCHAR(255)
	indoor_temperature	DECIMAL(5,1)
	indoor_humidity	INTEGER
	illumination	INTEGER
	co2_concentration	INTEGER
	pm2_5_concentration	INTEGER
	pm10_concentration	INTEGER
	formaldehyde_level	DECIMAL(4,3)
	noise_level	INTEGER
N	topic	VARCHAR(255)

图 1 数据表设计

#### 3.2 数据字典

1. id: 主键，由系统自动生成，唯一标识每条记录。
2. record\_time: 记录时间，日期格式为YYYY-MM-DD，此值为可变字符串类型，不能为空。
3. indoor\_temperature (REAL NOT NULL): 表示室内温度，单位通常为摄氏度(°C)。此值为实数类型，不能为空，用于记录环境的当前或历史温度状况。
4. indoor\_humidity (INTEGER NOT NULL): 表示室内湿度，单位通常为百分比(%)。此值为整数类型，不能为空，反映了空气中的水分含量，对评估室内舒适度至关重要。
5. illumination (INTEGER NOT NULL): 表示光照强度，单位通常为勒克斯(lux)。此值为整数类型，不能为空，用于衡量室内光线的明亮程度，影响能源使用和人的视觉舒适度。
6. co2\_concentration (INTEGER NOT NULL): 表示二氧化碳浓度，单位通常为ppm(百万分之一)。此值为整数类型，不能为空，用于监测室内空气质量，高浓度CO2可能影响人体健康和- 认知功能。
7. pm2\_5\_concentration (INTEGER NOT NULL): 表示空气中直径小于或

等于2.5微米的颗粒物（PM2.5）浓度，单位通常为 $\mu\text{g}/\text{m}^3$ 。此值为整数类型，不能为空，是评估空气质量、特别是- 对人体呼吸系统健康影响的重要指标。

8. pm10\_concentration (INTEGER NOT NULL): 表示空气中直径小于或等于10微米的颗粒物（PM10）浓度，单位通常为 $\mu\text{g}/\text{m}^3$ 。此值为整数类型，不能为空，反映较大颗粒物的污染水平，同样与空气质量及健康风险相关。
9. formaldehyde\_level (REAL NOT NULL): 表示甲醛浓度，单位通常为 $\text{mg}/\text{m}^3$ 。此值为实数类型，不能为空，甲醛是一种常见的室内污染物，过量暴露可引起眼鼻刺激、呼吸困难等症- 状。
10. noise\_level (INTEGER NOT NULL): 表示噪音水平，单位通常为分贝（dB）。此值为整数类型，不能为空，用于评估环境噪声，过高噪音会影响听力、睡眠质量和心理健康。

### 3.3 数据库介绍

本实验采用MySQL作为关系数据库。Mysql是一种关系型数据库管理系统，由瑞典 MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System, 关系数据库管理系统) 应用软件之一。

## 4 系统架构设计

### 4.1 系统总体架构

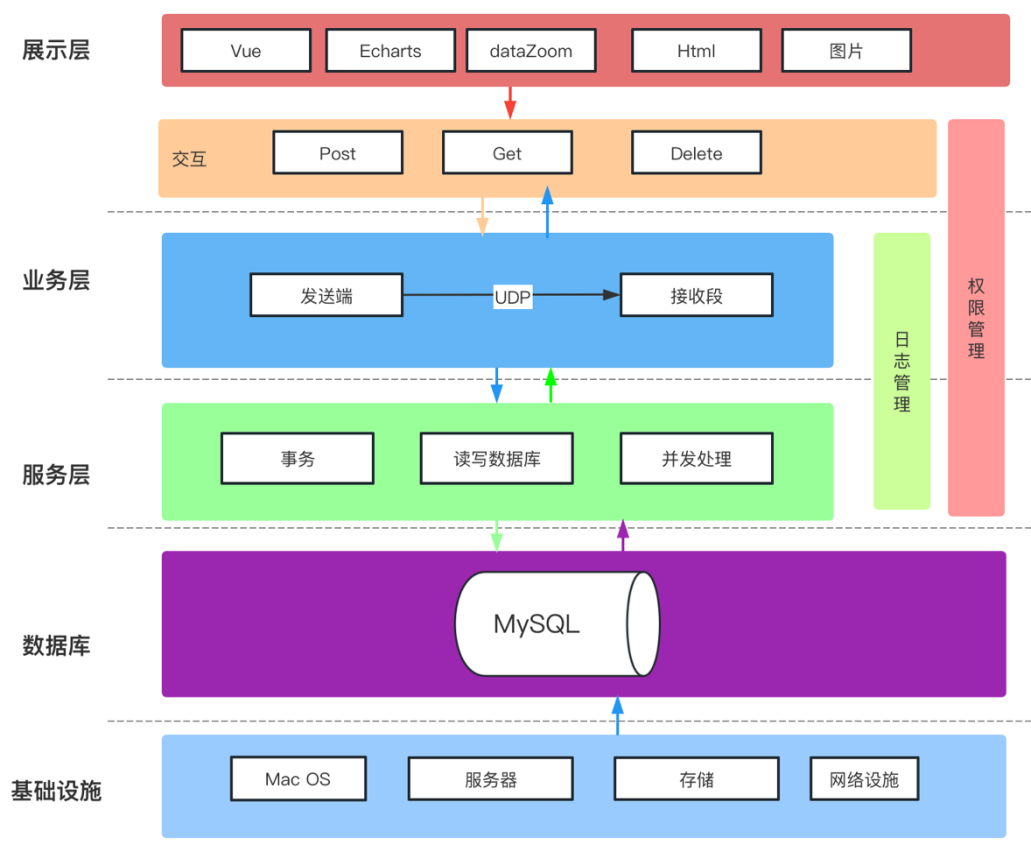


图 2 系统总体架构图

1. 前段采用Vue架构，使用html作为标记语言，使用Echarts和dataZoom作为数据可视化引擎。
2. 后端使用Springboot框架进行快速部署和开发，分为发送端和接收端，使用UDP协议进行通信。
3. 数据化持久层采用MySQL，支持事务和并发操作。



4.2 技术架构

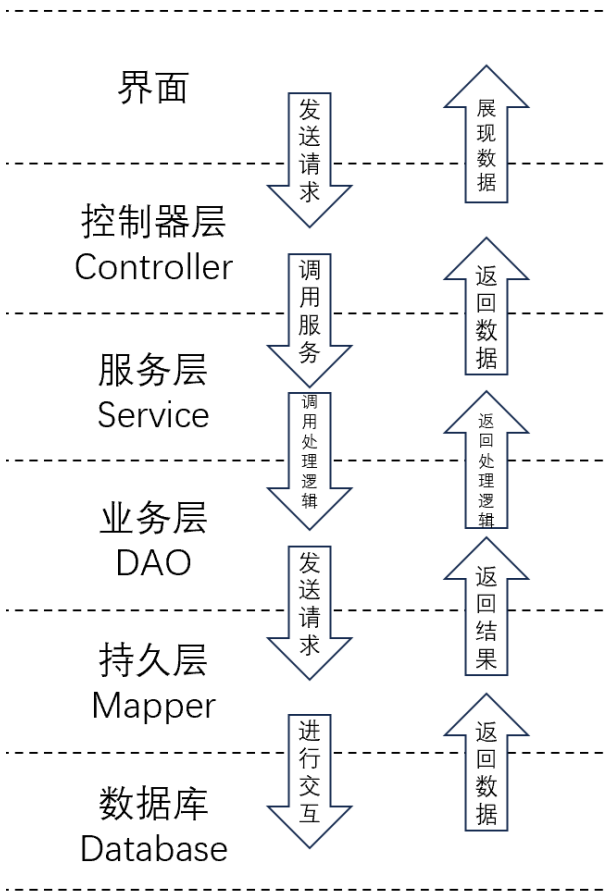


图 3 技术架构图

页面可以发送请求给控制器层，控制器层调用服务给服务层，服务层调用处理逻辑给业务层，业务层发送请求给持久层，持久层与数据库进行交互之后返回数据给持久层，持久层返回结果给业务层，业务层返回处理逻辑给服务层，服务层返回结果给控制器层，控制器层最后把展现的数据返回给页面。

## 5 接口定义

功能	类型	路径	描述
查找所有环境数据列表	Post	/system/data/list	需要登录，返回数据库中的所有环境数据
导出所有环境数据列表	Post	/system/data/export	需要登录，导出数据库中的所有环境数据
新增一条环境数据	Post	/system/data/add	需要登录，必须保证数据的合法性
根据 id 查找环境数据	Get	/system/data/edit/{id}	需要登录
跟句 id 修改环境数据	Post	/system/data/edit/{id}	需要登录
更新环境数据	Post	/system/data/edit	需要登录
删除一条环境数据	Delete	/system/data/remove	需要鉴权

表 1 接口定义及功能表

## 6 用户界面设计

### 6.1 主界面设计

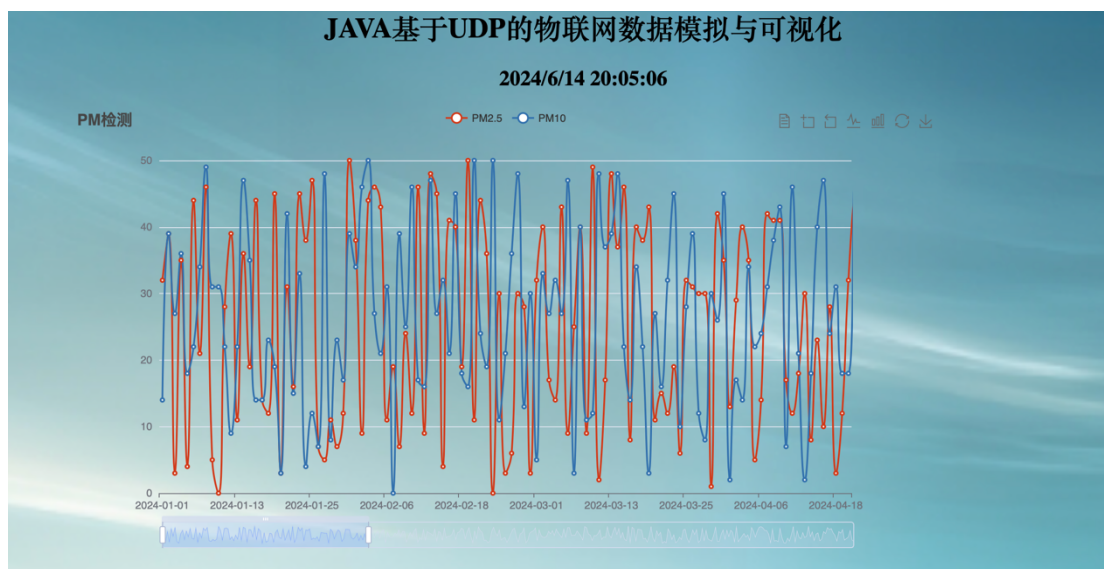


图 4 主界面设计

用户登录后即可进入主界面，主界面显示显示项目标题，下方显示当前时间，下方则是具体的数据展示，包括数据类型，数据按日变化折线图，横轴为日期，纵轴为具体数值，图表右上方显示不同操作按钮可对图表进行交互，分别是：展示数据列表、图表缩放、图表缩放重置、切换为折线图、切换为柱状图、刷新、保存。

## 6.2 图表缩放



图 5 图表放大图

使用鼠标框选即可将图表缩放，放大后的图片显示更小的日期范围内的数值变化。通过精细化时间的检测，我们能够更好看出来PM值受时间的波动趋势。

## 7 模块设计

### 7.1 UDPSend

模拟生成室内环境监测数据，并通过UDP协议实时发送至指定服务器。它循环遍历指定日期区间（2024年1月1日至2024年12月31日），每日生成一次包含室温、湿度、光照、CO2浓度、PM2.5浓度、PM10浓度、甲醛浓度及噪音水平的环境数据包，随后以JSON格式通过UDP发送至目标IP地址和端口。

#### 7.1.1 数据生成

generateEnvironmentData方法：

1. 输入：给定的日期时间LocalDateTime。
2. 过程：利用Random类生成一系列随机环境监测数值，确保数据的多样性和真实性。
3. 输出：一个EnvironmentData对象，封装了所有环境监测数据。

#### 7.1.2 UDP 数据传输

sendUdpData方法：

1. 过程：
  - 创建DatagramSocket实例准备发送UDP数据。
  - 将EnvironmentData对象转换为JSON字符串。
  - 封装JSON字符串为字节数组，并创建DatagramPacket。
  - 向指定的UDP\_IP和UDP\_PORT发送数据包。
2. 特性：采用try-with-resources确保DatagramSocket资源被正确关闭。

#### 7.1.3 主循环逻辑

1. 模拟周期：每日一次。
2. 控制逻辑：使用while循环遍历指定日期范围，每次迭代生成新的环境数据并发送。
3. 发送间隔：每次发送后，线程暂停1500毫秒，以控制发送频率并模拟实时数据流。

#### 7.1.4 异常处理

对于网络发送过程中的IOException，程序会捕获异常并打印错误信息至控制台，确保程序稳定性。

### 7.2 UDPreceive

作为一个UDP服务器，负责接收来自客户端的环境监测数据，并将接收到的数据通过HTTP POST请求转发至指定的Web服务器。程序具体功能及流程如下：

#### 7.2.1 UDP 数据接收

1. 启动服务：程序在本地开启UDP服务，监听端口10000，等待接收数据包。
2. 循环监听：进入无限循环，确保持续监听并处理所有到达的数据包。
3. 数据解析：每接收到一个数据包，解析出源IP地址、端口号及数据内容，并以友好的格式打印出来。

#### 7.2.2 数据转发至 Web 服务器

1. POST请求构造：接收到UDP数据后，立即构造一个HTTP POST请求，目标为预设的Web服务地址。
2. 请求体准备：将原始UDP数据作为请求体内容，指定内容类型为application/json; utf-8。
3. 发送请求：通过URLConnection发送POST请求至指定服务器，并处理响应。
4. 响应处理：检查HTTP响应码，成功时（状态码200-299），输出服务器响应内容；失败，则记录错误信息。

#### 7.2.3 技术实现

1. UDP通信：使用DatagramSocket类实现UDP数据的接收。
2. HTTP通信：通过URLConnection完成HTTP POST请求的发送与响应处理。
3. 字符编码：统一使用StandardCharsets.UTF\_8进行字符编码，确保数据传输的编码一致性。
4. 错误处理：采用try-catch结构，捕获并打印运行时异常，增强程序健壮性。

id	record_time	indoor_temperature	indoor_humidity	illumination	co2_concentration	pm2_5_concentration	pm10_concentration	formaldehyde_level	noise_level	topic
98	2024-04-07	17.4	63	824	418	42	31	0.044	45	(NULL)
99	2024-04-08	25.9	52	908	771	41	38	0.089	44	(NULL)
100	2024-04-09	19.5	41	749	753	41	43	0.092	44	(NULL)
101	2024-04-10	21.8	54	644	859	17	7	0.042	65	(NULL)
102	2024-04-11	19.5	64	370	607	12	46	0.074	38	(NULL)
103	2024-04-12	22.8	66	273	570	18	21	0.039	57	(NULL)
104	2024-04-13	24.3	50	712	429	30	2	0.067	34	(NULL)
105	2024-04-14	18.7	51	411	724	8	18	0.043	57	(NULL)
106	2024-04-15	19.8	52	882	779	23	40	0.029	50	(NULL)
107	2024-04-16	20.0	53	294	627	10	47	0.096	58	(NULL)
108	2024-04-17	17.7	55	920	492	28	24	0.083	33	(NULL)
109	2024-04-18	18.7	70	477	628	3	31	0.079	45	(NULL)
110	2024-04-19	20.4	50	347	696	12	18	0.074	67	(NULL)
111	2024-04-20	24.8	46	264	482	32	18	0.009	44	(NULL)
112	2024-04-21	24.6	46	284	642	44	47	0.044	58	(NULL)
113	2024-04-22	19.4	52	936	777	16	10	0.030	64	(NULL)
114	2024-04-23	18.8	65	634	626	29	49	0.098	40	(NULL)
115	2024-04-24	24.3	66	733	698	36	32	0.050	34	(NULL)
116	2024-04-25	22.3	43	505	500	14	8	0.004	38	(NULL)
117	2024-04-26	25.5	49	940	716	36	21	0.100	54	(NULL)
118	2024-04-27	16.0	46	300	464	40	46	0.043	39	(NULL)
119	2024-04-28	18.1	69	773	466	2	6	0.034	32	(NULL)
120	2024-04-29	20.1	51	556	780	33	46	0.084	63	(NULL)
121	2024-04-30	24.1	69	667	747	20	35	0.085	31	(NULL)
122	2024-05-01	19.9	51	611	431	17	48	0.016	60	(NULL)
123	2024-05-02	22.9	70	383	790	18	31	0.029	57	(NULL)
124	2024-05-03	25.9	69	758	716	30	8	0.093	61	(NULL)
125	2024-05-04	20.5	65	417	646	25	34	0.079	64	(NULL)
126	2024-05-05	22.3	55	754	713	40	25	0.083	68	(NULL)
127	2024-05-06	18.6	52	981	735	19	21	0.045	58	(NULL)
128	2024-05-07	16.5	59	553	417	36	23	0.098	70	(NULL)
129	2024-05-08	23.4	51	520	504	35	5	0.043	34	(NULL)
130	2024-05-09	16.0	70	797	650	3	48	0.084	42	(NULL)
131	2024-05-10	17.0	66	704	472	17	13	0.068	33	(NULL)
132	2024-05-11	23.8	63	312	756	32	1	0.089	40	(NULL)
133	2024-05-12	24.9	70	899	576	20	50	0.031	46	(NULL)

图 6 数据库详情图

7.3 数据可视化

通过Springboot快速启动CRUD，提供新增、查询、更新、删除等基本操作。前端通过接口请求environment\_data表中的数据。

从后端服务器获取环境监测数据，并对数据进行结构化处理，以便进行进一步分析或可视化展示。具体步骤如下：

1. 发起HTTP POST请求

使用axios.post方法向本地服务器地址http://127.0.0.1/system/data/list发送POST请求，异步获取数据列表。

await关键字确保等待请求完成后再继续执行后续代码。

2. 处理响应数据

通过response.data.rows访问API响应中的数据数组，并将其赋值给变量data。

3. 数据映射与拆分

对data数组中的每个元素执行映射操作，创建多个新数组，每个数组专注

于一种环境参数：

- dates: 记录时间列表。
- temperatureList: 室内温度列表。
- humidityList: 室内湿度列表。
- lightIntensityList: 光照强度列表。
- co2LevelList: 二氧化碳浓度列表。
- pm25ConcentrationList: PM2.5浓度列表。
- pm10ConcentrationList: PM10浓度列表。
- formaldehydeLevelList: 甲醛浓度列表。

将处理后的数据通过Echarts可视化展示。

## 8 备注

### 8.1 实验人员

顾畅：22920212204375，构建编码、测试、文档撰写

陈新：22920212204359，设计建模，文档编写

### 8.2 Git 仓库

仓库地址：<https://github.com/ChangGu328/IOTProject.git>