



华侨大学
HUAQIAO UNIVERSITY

本科毕业论文（设计）

题目： 面向养老监护的毫米波雷达
人体运动理解系统开发

院（系） 计算机科学与技术学院
专 业 软件工程
届 别 2026 届
学 号 2225122007
姓 名 法俊良
指导老师 王华珍 副教授

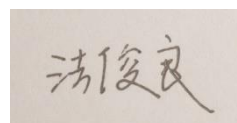
华侨大学教务处印制

2026 年

华侨大学本科毕业论文（设计）诚信承诺书

本人（姓名）法俊良学号2225122007专业
郑重承诺：所呈交的毕业论文（设计）（题目：面向养老监护的毫米波雷达人体运动理解系统开发）是在指导教师指导下自主完成，毕业论文（设计）选题和研究内容中不存在不正当引用、抄袭、伪造、篡改、代写、买卖等行为，如有违规行为，本人愿意承担一切责任。

承诺人（签名）：

A rectangular box containing a handwritten signature in black ink, which reads '法俊良' (Fa Junliang).

2026 年 6 月 6 日

摘要

随着人口老龄化程度持续加深，面向居家养老、病房陪护与慢病监测场景的连续健康监护需求日益增长。传统接触式监测方案通常依赖穿戴设备，存在用户依从性不足、监测连续性不强、隐私保护体验一般等问题。毫米波雷达具备非接触、隐私友好、弱光环境适应性强和可连续部署等特点，为生命体征感知与人体状态识别提供了新的技术路径

针对传统接触式监测方案依从性不足、连续性不强的问题，构建了基于毫米波雷达与边缘控制器的感知与传输链路。系统底层以 LD6002 系列毫米波雷达为感知源，采用 ESP32-S3 作为边缘控制器，实现雷达串口报文解析、跌倒状态识别、有人 / 无人状态提取、WiFi 联网、SoftAP 配网与 MQTT 数据上报，完成监护数据的边缘感知与可靠传输，解决传统方案在非接触、连续性部署方面的不足。

针对现有养老监护系统缺乏适配毫米波雷达时序特性的数据接入与存储方案的问题，本文提出了一种面向监护场景的多组件协同时序数据处理架构。首先，基于 Mosquitto 构建多设备并发数据接入层，针对居家环境下雷达数据高频、易丢包的特点，优化了 MQTT 消息分发与缓存策略，保障多设备监护数据的稳定接收；其次，基于 InfluxDB 设计适配生命体征与状态数据的时序存储模型，针对心率、呼吸、跌倒状态等监护数据的特性定制存储策略，实现了雷达数据的高效时序存储与低延迟查询；以上架构为后续监护业务功能开发提供了高可靠的数据基础。

针对现有养老监护系统缺乏适配业务需求的可视化展示与管理载体的问题，本文基于 Grafana 开展养老监护场景业务化适配重构，面向多设备管控与风险预警核心业务需求，定制构建设备管理、卡片管理、风险弹窗告警、设备在线检测及分页筛选等场景专属管控业务模块。在真实居家场景数据的测试结果表明该系统能够完成跌倒状态与人体在位状态的稳定上报，验证了毫米波雷达、物联网时序数据平台与养老监护业务系统结合的可行性，为智慧养老健康监护系统的工程落地提供了参考。

关键词：毫米波雷达；智慧养老；人体运动理解；InfluxDB；Grafana

ABSTRACT

As the population continues to age, the demand for continuous health monitoring in home care, hospital ward care, and chronic disease monitoring scenarios is growing. Traditional contact-based monitoring solutions typically rely on wearable devices, which suffer from issues such as low user compliance, poor monitoring continuity, and inadequate privacy protection. Millimeter-wave radar offers non-contact operation, privacy respect, strong adaptability in low-light environments, and the ability for continuous deployment, providing a new technological pathway for vital sign sensing and human condition recognition.

To address the issues of insufficient compliance and weak continuity in traditional contact-based monitoring solutions, a perception and transmission link based on millimeter-wave radar and edge controllers was designed and implemented. At the core of the system, the LD6002 series millimeter-wave radar is selected as the sensing source, while the ESP32-S3 is used as the edge controller. This configuration makes it possible to implement radar serial message parsing, fall detection, presence status extraction, Wi-Fi connectivity, SoftAP network configuration, and MQTT data reporting. In this way, reliable edge-side sensing and transmission of monitoring data can be achieved, while also addressing the limitations of traditional solutions in non-contact and continuous deployment scenarios.

To address the lack of data acquisition and storage solutions adapted to the temporal characteristics of millimeter-wave radar in existing elderly care monitoring systems, this paper further proposes a multi-component collaborative time-series data processing architecture tailored for monitoring scenarios. First, a multi-device concurrent data acquisition layer is constructed based on Mosquitto. Considering the high-frequency characteristics of radar data as well as the risk of packet loss in home environments, MQTT message distribution and caching strategies are optimized to ensure stable reception of monitoring data from multiple devices. Second, a time-series storage model suitable for vital signs and status data is designed based on InfluxDB.

Customized storage strategies are implemented for monitoring data such as heart rate, respiration, and fall status, thereby enabling efficient time-series storage and low-latency querying of radar data. This architecture provides a highly reliable data foundation for the subsequent implementation of monitoring business functions.

Addressing the lack of visualization and management tools tailored to business requirements in existing elderly care monitoring systems, this paper carries out a business-oriented reconstruction of elderly care monitoring scenarios based on Grafana. Focusing on the core business needs of multi-device management and risk early warning, several scenario-based control modules are custom developed, including device management, dashboard management, risk pop-up alerts, device online detection, as well as paginated filtering functions. Test results based on real home environment data show that the system can stably report fall events together with resident presence status. This validates the feasibility of integrating millimeter-wave radar with an IoT time-series data platform and elderly care monitoring systems, providing a reference for the engineering implementation of smart elderly care and health monitoring systems.

Keywords: Millimeter-wave radar; Smart elderly care; Human motion understanding; InfluxDB; Grafana

目录

摘要	IV
Abstract.....	VII
第一章 绪论.....	1
1.1 研究背景	1
1.2 研究意义	2
1.2.1 理论意义	2
1.2.2 技术意义	3
1.2.3 应用意义	3
1.3 国内外研究现状	4
1.3.1 毫米波雷达生命体征与人体状态感知研究	4
1.3.2 物联网数据传输与时序存储研究	5
1.3.3 监护业务可视化与系统集成研究	5
1.3.4 研究空白与本文切入点	5
1.4 研究内容	6
1.5 论文结构	7
第二章 系统需求分析.....	9
2.1 业务场景分析.....	9
2.1.1 场景描述	9
2.1.2 典型用户角色	9
2.2 功能需求分析.....	10
2.2.1 设备端感知需求.....	10
2.2.2 云端数据接入与存储需求.....	11
2.2.3 业务门户需求	11
2.2.4 设备管理与卡片管理需求.....	12
2.2.5 监护数据服务需求	13
2.3 非功能需求分析	13
2.3.1 实时性需求	13
2.3.2 可靠性需求	13
2.3.3 易维护性需求	13
2.3.4 可扩展性需求	14
2.4 可行性分析.....	14

2.4.1 技术可行性	14
2.4.2 经济可行性	14
2.4.3 运行可行性	14
2.5 本章小结	14
第三章 系统概要设计	15
3.1. 系统总体架构设计	15
3.2. 系统数据流设计	16
3.3 功能模块设计	17
3.3.1 边缘感知模块	17
3.3.2 数据接入与存储模块	17
3.3.3 健康监测模块	18
3.3.4 设备管理模块	18
3.3.5 卡片管理模块	18
3.3.6 监护数据服务模块	19
3.4 数据模型设计	19
3.4.1 设备配置模型	19
3.4.2 门户卡片配置模型	19
3.4.3 时序数据模型	19
3.5 接口设计	20
3.5.1 设备管理接口	20
3.5.2 门户卡片接口	20
3.5.3 InfluxDB 查询代理接口	21
3.6 监护数据服务设计	21
3.7 本章小结	22
第四章 系统详细设计与实现	23
4.1 基于毫米波雷达的感知与传输链路实现	23
4.1.1 硬件组成与通信方式	23
4.1.2 串口报文解析	23
4.1.3 WiFi 连接与 SoftAP 配网	24
4.1.4 MQTT 数据上报	25
4.2 面向监护场景的多组件协同时序数据处理架构实现	26
4.2.1 订阅流程	26
4.2.2 时序字段映射	27
4.2.3 风险检测与短信告警	28

4.3 基于 Grafana 的养老监护业务化适配重构	29
4.3.1 业务门户重构与状态组织.....	30
4.3.2 多类型卡片渲染策略	30
4.3.3 分页、筛选与手动刷新	32
4.3.4 跌倒风险弹窗机制	32
4.4 设备管理模块实现	33
4.4.1 设备数据模型	33
4.4.2 设备管理 API.....	33
4.4.3 前端设备管理弹窗	34
4.5 卡片管理模块实现	35
4.5.1 卡片配置模型	35
4.5.2 卡片管理逻辑	36
4.6 InfluxDB 缓存代理与回放机制实现	37
4.6.1 查询代理设计	37
4.6.2 固定时段回放与实时合并.....	38
4.7 监护数据服务实现	39
4.7.1 查询服务目录组织.....	39
4.7.2 时间窗口与设备映射处理.....	39
4.7.3 查询参数与结果组织规则.....	40
4.8 关键实现代码说明	41
4.8.1 设备端上报结构.....	41
4.8.2 门户卡片绑定逻辑	42
4.8.3 设备在线状态检查	43
4.9 本章小结	44
第五章 系统测试	45
5.1 测试环境.....	45
5.2 功能测试方案	46
5.2.1 设备端功能测试.....	46
5.2.2 云端入库测试	46
5.2.3 业务门户测试	46
5.2.4 管理模块测试	46
5.2.5 监护数据服务测试	47
5.3 关键测试用例.....	47
5.3.1 跌倒状态上报测试.....	47

5.3.2 设备离线检测测试	47
5.3.3 设备配置一致性测试	48
5.3.4 卡片配置联动测试	48
5.4 测试结果分析	49
5.4.1 功能闭环分析	49
5.4.2 链路稳定性分析	50
5.4.3 系统局限性与改进方向	50
5. 本章小结	50
第六章 总结与展望	51
6.1 论文总结	51
6.2 未来展望	52
参考文献	55
致谢	59

第一章 绪论

1.1 研究背景

随着我国老龄化社会进程的不断推进，老年人跌倒、长期卧床、夜间独居以及慢性病并发等风险，已经成为居家养老以及智慧医疗当中的重点问题。传统监测方式主要依靠人工巡检、穿戴式手环、床旁接触式传感器以及视频摄像头来开展监测工作。这些方案虽然能够获取一定的健康数据，但同时也存在较为明显的不足：其一，穿戴式设备需要老年人长期进行佩戴，整体依从性相对有限；其二，摄像头方案在家庭以及病房环境当中，比较容易引发隐私方面的顾虑；其三，接触式设备通常不利于开展长时间、舒适以及无感知的连续监测；其四，在多设备场景下，数据集成以及展示方面的复杂度也会相对较高。毫米波雷达利用电磁波反射获取目标距离、运动状态与微动特征，尤其适合生命体征监测和粗粒度行为理解场景。基于 60GH 频段的毫米波雷达能够感知呼吸、心率、体动幅度与距离变化，在夜间、遮挡或光照变化场景下仍可保持较高稳定性^[1-3]。与视频监控相比，毫米波雷达不直接采集人体图像，更符合居家养老中“连续监测但弱侵扰”的场景需求。

近年来，智慧养老监护体系正由传统单点式数据采集，向端边云协同、全链路数据治理的平台化架构转型。随着物联网通信协议、时序数据库技术与可视化门户架构的日趋成熟，监护系统突破单一数据采集局限，具备了多终端数据汇聚、时序化存储、智能风险研判、历史数据回溯与设备运维管控的完整能力，为构建感知、传输、存储、展示、管理一体化的端到端监护闭环提供了坚实技术支撑。

立足现有监测方案的场景短板与行业技术发展趋势，本文面向智慧养老实际业务需求，提出一套毫米波雷达驱动的端到端人体健康与行为监护系统架构。系统底层依托毫米波雷达与嵌入式边缘控制器，完成人体生命体征及行为状态的非接触精准感知，为全链路监护体系提供源头数据支撑；中间层基于 MQTT 消息协议与 InfluxDB 时序数据库，实现多设备并发接入、监测数据可靠传输与时序化统一管理，承担全系统数据中转与治理核心作用；上层以 Grafana 可视化架构为载体，结合自研业务逻辑与监护数据服务能力，承载健康监测门户展示、设备运

维管控、状态智能分析及风险联动预警等核心业务，由此将底层传感器感知、中间层时序数据治理与上层监护业务分析有机融合，构建形成完整的端到端健康监护闭环。整套方案兼顾场景适配性、工程落地性与长期运维稳定性，有效弥补了传统监护系统碎片化、隐私性弱、连续性不足的缺陷，可为居家养老、病房陪护等场景的智能化健康监护工程落地提供理论参考与可复用的实践范式。

1.2 研究意义

1.2.1 理论意义

本研究并非局限于理想实验环境下单项指标可检测性的论证，而是面向智慧养老监护的实际应用过程，重点分析毫米波雷达感知数据在监护业务场景中的组织、解释与应用方式。相较于仅关注单一生命体征识别准确率的研究思路，本文进一步关注雷达采集到的原始数据如何经过传输、解析、存储与业务映射，最终转化为能够表征人体存在、跌倒风险及生命体征变化的监护语义。

围绕这一问题，系统构建了一套自底向上的完整架构。底层负责人体感知与数据采集，中间层承担时序数据治理与流转，上层则面向监护业务完成状态分析与风险表达。换句话说，研究重点已经不再只是“检测到了什么”，而是进一步关注“这些数据如何真正服务于监护行为”。

某种程度上，这种贯通式设计弥补了当前非接触式健康监护研究里一个比较常见的断层：很多系统结构看上去完整，但数据到了业务层后，语义表达往往是割裂的。传感器输出是一套逻辑，监护界面又是另一套逻辑，中间缺少真正连贯的数据解释过程。本文尝试把这部分链路重新打通，让数据从采集源头开始，就逐渐向“人体状态理解”靠近，而不是停留在孤立指标展示层面。

由此形成的研究结论，对于智慧健康监测系统如何组织架构、如何设计数据流转路径，以及如何建立监护业务映射关系，都提供了一定参考价值。同时，也在一定程度上扩展了毫米波雷达技术在养老监护场景中的应用边界。

1.2.2 技术意义

在实际工程开发过程中，一个很常见的问题是：模块虽然都能运行，但彼此之间缺乏协同。设备端是一套逻辑，云端是一套逻辑，业务层又是另一套逻辑，最后系统容易出现功能分散、状态不同步的问题。针对这一情况，本文在技术实现上没有采用“功能堆叠”的方式，而是更强调整体联动。

设备侧部分，系统基于毫米波雷达与嵌入式控制器，完成了从雷达报文解析、人体存在识别到跌倒检测的一体化开发。同时，将 SoftAP 配网、MQTT 数据上传等功能放进统一框架中处理，避免设备能力被拆成彼此独立的模块。这样做虽然增加了前期整合复杂度，但后续维护会稳定很多。

云端部分，则围绕 MQTT 消息机制与 InfluxDB 时序数据库，对多设备接入、数据写入、缓存回放等流程进行了统一管理。除此之外，系统还结合 Grafana 的可视化能力，对监护门户、风险提示弹窗以及业务展示逻辑进行了针对性重构，使其更贴近养老监护场景下的实际操作习惯，而不是停留在通用仪表板展示层面。

服务层进一步通过统一查询接口，将心率、呼吸频率、跌倒风险等状态信息进行整合调用，实现页面联动与状态同步。这样一来，监护页面中的数据刷新、状态切换以及风险提示，不再依赖多个孤立模块分别处理，而是形成了一套相对统一的数据协同机制。

整体来看，该方案在嵌入式开发、时序数据治理、可视化系统重构以及后端服务协同等方面进行了系统性验证，形成了一套可复用的非接触式养老监护技术实现路径，对同类型系统的工程开发具有直接的参考价值。

1.2.3 应用意义

基于上述设计与实现，本研究构建的非接触式养老监护系统能够较好地适应居家养老、病房照护以及康复监测等多种应用环境，针对传统监护方式中常见的依从性不足、隐私暴露风险高、使用流程繁琐等问题给出了现实可行的解决方案。

系统集成跌倒风险提示、人体在位状态识别、多设备集中管理以及监护数据可视化展示等功能，配合图表与页面跳转机制，为护理人员、老年人家属及普通用户提供了相对直观的使用入口，也在一定程度上降低了日常监护的操作门槛。

1.3 国内外研究现状

1.3.1 毫米波雷达生命体征与人体状态感知研究

在毫米波雷达生命体征检测与人体状态感知领域,国内外学者已开展了较多研究,研究内容从早期的呼吸、心率等基础生命体征检测,逐步扩展到人体存在判断、床旁状态监测、睡眠姿态识别、跌倒检测和人体活动识别等多个方向。毫米波雷达具有非接触、弱光环境适应性强、隐私保护性较好等特点,能够在不依赖摄像头和穿戴设备的情况下获取人体微动、距离和状态变化信息,因此在智慧养老、病房陪护和居家健康监测等场景中具有较高的应用价值。

Ma 等在室内人体活动与生命体征监测研究中,分析了毫米波雷达在人体定位、跌倒检测以及呼吸、心跳检测等任务中的应用可行性,验证了毫米波雷达在室内人体感知场景中的实用性^[1]。Jang 等针对床旁人员监测问题开展研究,将毫米波雷达用于检测人在床状态变化和呼吸率估计,实验场景覆盖卧床、离床以及床旁异常变化等情况,说明毫米波雷达在卧床监护和夜间看护场景中具有较好的应用前景^[2]。Zheng 等则从生命体征信号提取角度出发,研究了基于多距离单元的呼吸和心率信号提取方法,通过利用 FMCW 毫米波雷达的距离分辨能力改善复杂环境下生命体征信号的信噪比,提高了雷达生命体征检测的稳健性^[3]。

除基础生命体征检测外,毫米波雷达在人体状态识别方面的研究也在不断扩展。Liu 等围绕睡眠姿态识别问题,利用毫米波雷达获取人体睡眠状态下的姿态信息,并通过多视角特征融合方式提升复杂睡眠环境下的识别效果,为非接触式睡眠监测提供了参考^[4]。Tang 等进一步面向毫米波雷达人体姿态估计任务,针对雷达点云稀疏、噪声较多和几何特征不稳定等问题,引入几何感知特征学习方法,使毫米波雷达的人体感知,从简单状态判断逐步拓展到姿态估计和运动语义理解层面^[6]。

综合来看,该方向已在算法层面形成丰富积累,但研究重心普遍偏向算法精度与单一指标的提取,缺少与上层业务系统的工程化耦合。受制于此,毫米波雷达在“数据落地到业务页面”这一关键环节上仍存在较为明显的实践断层,这正是本文需要正视的第一类研究空白。

1.3.2 物联网数据传输与时序存储研究

在物联网数据传输与平台集成方面，MQTT 以轻量、低开销、适合边缘终端的特性被广泛采用。相关研究指出，通过消息队列机制可以实现端云之间高效且可扩展的数据同步^[7-9]。InfluxDB 作为典型时序数据库，适合保存带时间戳的健康指标数据，并支持聚合、窗口计算和快速可视化对接^[10-12]。

但需指出的是，上述研究主要面向通用 IoT 场景或单点应用，未针对毫米波雷达高频、字段维度多、易丢包等数据特征系统设计接入策略与存储模型；面向养老监护业务的字段语义（如跌倒持续时间、人体在位、有效帧时间等）也较少被纳入存储层考虑。这一断层构成了本文第二类研究空白：缺乏针对毫米波雷达数据特性的多组件协同时序数据处理架构。

1.3.3 监护业务可视化与系统集成研究

在监护业务系统方面，当前研究重点逐渐从单纯数据存储转向状态组织、风险提示和页面联动^[10,12]。现阶段许多研究聚焦于单一指标的可视化展示，但围绕“嵌入式设备实时监测数据—统一查询接口—业务页面反馈”的工程落地工作仍相对有限。特别是在智慧养老健康监测场景中，如何将心率、呼吸、跌倒状态等时间序列指标组织为稳定可用的服务接口，并形成可靠、简洁、面向场景的监护结果，仍有很大研究空间。

从系统集成层面看，国内外面向智慧养老的端到端系统研究多偏重单一功能模块（例如仅跌倒识别、仅心率监测）或单一展示页面，缺乏从感知—传输—存储—展示到业务服务的完整闭环。这构成了本文第三类研究空白：缺乏针对养老监护业务需求深度适配的工程化实践验证。

1.3.4 研究空白与本文切入点

综合上述分析，可以将现有研究的不足归纳为以下三点：一是在系统形态上，现有工作多停留在单点算法或单点展示层面，缺少面向居家养老场景的端到端感知—传输—存储—展示一体化系统实现方案；二是在数据组织上，已有 IoT 平台并未围绕毫米波雷达高频字段、状态语义和有效性标记等特点设计接入与存储策

略，缺少面向监护场景的多组件协同时序数据处理架构；三是在业务支撑上，现有监护可视化方案多采用通用 Dashboard 直接展示，缺少对多设备管理、风险弹窗、在线检测与配置化管理等业务需求的深度适配与工程化实践验证。基于上述判断，本文研究围绕养老监护业务需求开展架构设计、模块定制与系统集成，最终形成一套可用于家庭与机构场景的连续健康监护系统。

1.4 研究内容

本文围绕面向养老监护的毫米波雷达人体运动理解系统展开研究，主要包括以下几个方面：

（1）毫米波雷达感知与传输链路构建

针对传统接触式监测方案存在依从性不足以及连续性不强的问题，本文设计并实现了基于毫米波雷达以及 ESP32-S3 边缘控制器的感知与传输链路。该链路在硬件层面选用 LD6002 系列毫米波雷达来完成跌倒状态、人体存在状态以及生命体征相关字段的采集工作；在固件层面，基于 FreeRTOS 任务模型实现了串口报文解析、状态机管理、WiFi 联网、SoftAP 配网以及 MQTT 周期上报等功能；在传输策略方面，仅对门户展示以及业务判断所需要的关键字段进行上报，从而在保证连续性的同时，降低消息体长度以及网络负载。

（2）多组件协同时序数据处理架构搭建

针对现有养老监护系统缺乏适配毫米波雷达时序特性的数据接入与存储方案的问题，本文提出一种面向监护场景的多组件协同时序数据处理架构。在接入层，基于 Mosquitto 构建多设备并发数据接入口，结合居家环境下雷达数据高频、易丢包、状态切换密集的特点，配置主题分发、QoS 与缓存策略；在存储层，基于 InfluxDB 设计以 device_data 为统一测量、以设备标识与主题为标签、以心率/呼吸/距离/体动/跌倒/在位等字段为时序字段的存储模型；在数据组织层，结合标签与字段的语义差异制定面向监护业务的查询模式与回放策略，为上层业务功能开发提供高可靠的数据基础。

（3）基于 Grafana 的养老监护业务化适配重构

针对现有养老监护系统缺乏适配业务需求的可视化展示与管理载体的问题，本文在 Grafana 框架基础上开展养老监护场景的业务化适配重构。面向多设备管控与风险预警的核心业务需求，本文定制构建了业务门户、多类型卡片渲染、设备管理、卡片管理、风险弹窗告警、设备在线检测、分页筛选等场景专属模块，并通过统一的监护数据服务接口将底层时序数据组织为前端可直接消费的结构化结果，从而打通从底层感知到上层业务结果的完整链路。

1.5 论文结构

本文共分为六章。整体结构与各章主要内容如下：

第一章绪论：阐述课题研究背景、研究意义、国内外研究现状与三类研究空白，给出本文主要研究内容与组织结构；

第二章系统需求分析：在业务场景分析的基础上，系统梳理设备端、平台端、门户监护数据服务的功能需求与非功能需求，并对技术、经济与运行可行性进行分析；

第三章系统总体设计：给出系统的总体架构、数据流、模块划分、数据模型与接口设计，并对监护数据服务层的总体定位进行说明；

第四章系统详细设计与实现：详细阐述毫米波雷达感知与传输链路、多组件协同时序数据处理架构以及基于 Grafana 的业务化适配重构的设计与实现；

第五章系统测试：给出测试环境、功能测试用例、关键测试用例与测试结果分析，验证系统在连续监护场景下的工程可用性；

第六章总结与展望：对全文工作进行总结，并就后续可拓展的研究方向给出展望。

第二章 系统需求分析

本章在前述研究背景与三类研究空白的基础上，对本系统的应用场景、用户角色、功能需求与非功能需求进行系统化梳理，并对系统的可行性进行分析，为后续总体设计与详细实现提供约束依据。

2.1 业务场景分析

2.1.1 场景描述

本系统面向智慧养老健康监测场景，重点服务于独居老人、家庭看护者、护理人员以及养老机构管理人员等典型角色。系统部署形态可概括为“房间级毫米波雷达采集+云端平台集中汇聚+业务化门户与监护服务支撑”三段式结构：在每一个被监护的物理空间（卧室、卫生间、客厅、病房床旁等）内部署一台毫米波雷达设备及其边缘控制器，由控制器完成本地感知与上行汇聚；云端通过 MQTT 消息中间件接收并存储设备上报的数据；业务侧通过定制化的门户与服务接口完成多设备状态展示、风险联动与配置化管理。

从业务运行节奏看，该场景具有以下显著特征：第一，部署规模可弹性扩展，单户家庭通常需要 1-3 台设备，养老机构则可能涉及数十至上百台设备的并行接入，对系统的多设备承载能力提出明确要求；第二，风险事件具有强时效性，跌倒、长时间无人活动、心率/呼吸严重异常等情况需要在数秒级内被识别并反馈，对端到端时延提出约束；第三，使用人群以非专业用户为主，要求页面表达简洁直观、操作路径短；第四，监护任务具有显著的连续性属性，要求系统具备长期稳定运行与短时异常自恢复能力。

2.1.2 典型用户角色

结合上述场景，本系统识别出以下典型用户角色及其核心诉求：

（1）专业护理人员：通过简单的界面快速判断被监护者当前状态，并在异常情况下及时收到提醒；同时可访问业务门户监测、风险告警、指标查询、设备

在线状态、设备管理、门户卡片管理和仪表盘跳转等全部业务模块，根据系统提供的健康数据时间序列开展面向多房间、多设备场景的连续照护与批量管控。

(2) 家属/家庭看护者：该角色主要对应居家养老场景中的亲属照护需求，存在的必要性在于家庭成员通常不是专业护理人员，但需要及时了解老人所处房间的在位状态、心率/呼吸等基础监测结果以及跌倒风险提示。与护理人员相比，家属侧重于查看健康监测业务门户、接收风险告警、查询关键生命体征和跳转查看仪表盘，不直接承担设备信息维护、门户卡片配置等管理操作。

(3) 机构管理者：通过系统进行多设备配置化管理，降低长期运维成本。

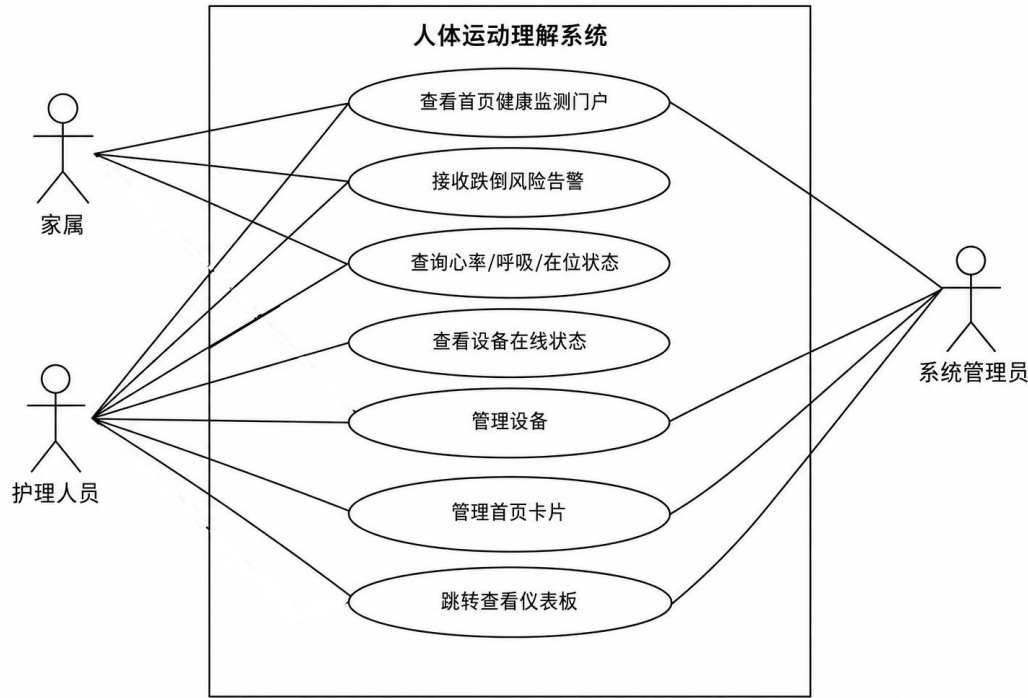


图 2-1 系统业务用例图

2.2 功能需求分析

2.2.1 设备端感知需求

设备端作为整条链路的起点，承担着原始信号采集、状态识别与可靠上报的职责，其需求具体包括：

- (1) 通过 UART 接口读取毫米波雷达输出的报文数据；
- (2) 解析跌倒状态帧与人体存在状态帧；

- (3) 根据系统启动情况持续统计跌倒持续时间；
- (4) 通过 WiFi 接入网络，并在配网失败时进入 SoftAP 配网模式；
- (5) 周期性将设备编号、时间戳、跌倒状态、跌倒计时和人体存在状态封装为 JSON 后发布到 MQTT 主题。

2.2.2 云端数据接入与存储需求

云端需要承担消息转发、数据清洗、时序入库和查询支持等职责，具体包括：

- (1) 订阅设备端发布的 MQTT 消息；
- (2) 将 JSON 或原始十六进制数据解析为结构化字段；
- (3) 使用统一测点名称保存到 InfluxDB 中；
- (4) 为心率、呼吸、距离、体动、跌倒、人存在等字段提供查询支持；
- (5) 支持缓存、固定时段回放与实时数据合并。

2.2.3 业务门户需求

基于 Grafana 进行养老监护业务化适配重构，形成面向养老监测场景的专用门户。该门户需满足以下需求：

- (1) 以卡片形式展示多个房间或设备的核心指标；
- (2) 支持心率设备、血氧设备和跌倒检测设备三类卡片；
- (3) 支持按设备类型筛选、分页浏览和手动刷新；
- (4) 支持告警弹窗与风险确认；
- (5) 支持点击卡片跳转到对应仪表板；
- (6) 支持设备在线状态检测与展示。

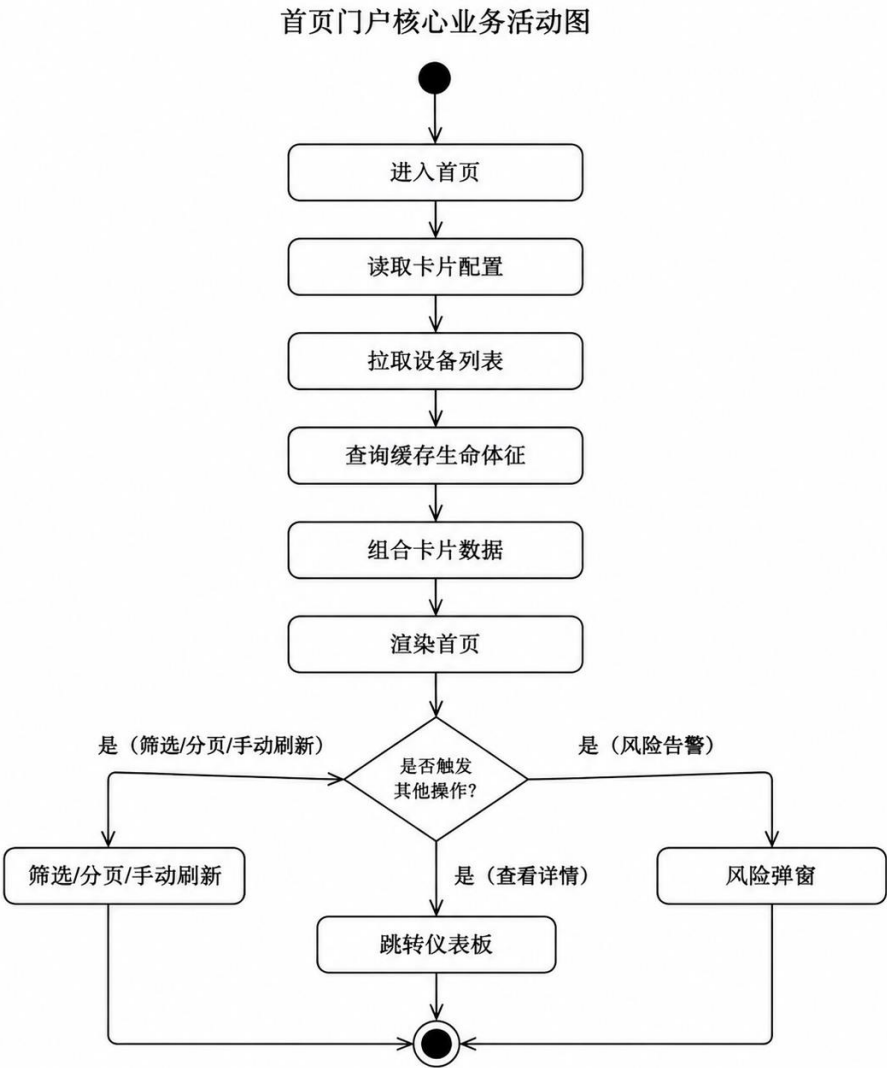


图 2-2 业务门户核心业务活动图

2.2.4 设备管理与卡片管理需求

由于系统支持多设备部署，因此门户配置必须可维护。具体需求如下：

- (1) 支持设备的新增、编辑、删除和列表查看；
- (2) 支持记录设备名称、MAC 地址、类型和备注；
- (3) 支持门户卡片与设备的一对一绑定；
- (4) 支持卡片与 Dashboard UID 的绑定；
- (5) 支持变更卡片顺序和房间名称。

2.2.5 监护数据服务需求

为适配养老监护场景的业务逻辑与数据规范，在业务门户系统中通过定制化参数配置、查询规则适配与业务逻辑调校，实现面向监护业务的数据分析与服务能力，包括：

- (1) 对指定时间段的心率、呼吸率、跌倒状态和人体在位状态进行查询与汇总；
- (2) 根据房间、设备或卡片配置对监护数据进行分类展示与历史回放；
- (3) 在检测到跌倒持续、设备离线或数据异常时触发风险提示与状态标记；
- (4) 为业务门户、设备管理和卡片管理提供统一的数据服务接口；
- (5) 支持监护数据的趋势分析、页面联动以及连续运行场景下的稳定展示。

2.3 非功能需求分析

2.3.1 实时性需求

跌倒和人体状态属于强时效性信息，系统需要实现秒级上报与刷新。设备端的 MQTT 发布周期设置为 1 秒，业务门户定时刷新与缓存更新机制也围绕秒级数据展开，满足实时监控需求。

2.3.2 可靠性需求

系统必须能够在网络波动和短时中断场景下继续运行，因此设备端需要具备自动重连、默认 WiFi 配置和 SoftAP 配网能力；云端需要具备健康检查、容器化部署、缓存机制与数据回放能力。

2.3.3 易维护性需求

在多设备场景下，若设备信息与卡片信息均通过硬编码方式维护，将显著增加系统运维复杂度。因此，系统引入设备表与门户卡片表，将配置过程转化为后台可管理操作。

2.3.4 可扩展性需求

系统不仅要支持当前的跌倒检测和生命体征监测，还应为后续扩展更多传感器类型、更多 Dashboard 以及更多监护数据服务能力留下空间。设备类型字段、卡片绑定字段和查询服务接口均体现了这种设计。

2.4 可行性分析

2.4.1 技术可行性

毫米波雷达、ESP32、MQTT、InfluxDB 与 Grafana 均属于较为成熟的技术组件，具备良好的工程适配性与集成基础。系统在 Grafana 框架基础上完成业务化适配重构，并通过 Go 语言补充 REST API、结合 Python 脚本实现消息处理与风险分析，因此该技术路线具有较好的可行性。

2.4.2 经济可行性

系统核心硬件采用毫米波雷达模组与通用嵌入式控制器，成本相对可控；软件端主要使用开源组件，部署方式以 Docker 容器为主，降低了环境运维难度。

2.4.3 运行可行性

系统采用前后端分层和容器化部署方式，既适合实验室环境验证，也适合家庭或病房场景部署。门户管理能力的加入进一步降低了运行维护门槛。

2.5 本章小结

本章从业务场景、用户角色出发，围绕设备端感知、数据接入与存储、业务门户、配置管理与监护数据服务五个层面对系统功能需求进行了梳理，并就实时性、可靠性、易维护性、可扩展性与安全隐私等非功能需求做了系统性约束。本章所归纳的需求项与可行性结论，为后续章节的总体架构设计与详细实现提供了直接依据。

第三章 系统概要设计

基于第二章的需求分析结论，本章从总体架构、数据流、功能模块、数据模型、接口设计以及监护数据服务六个角度，对面向养老监护的毫米波雷达人体运动理解系统进行总体设计，作为第四章详细实现的设计依据。

3.1. 系统总体架构设计

本文研究遵循三条主线：以毫米波雷达与边缘控制器为感知与传输入口，构建端到端非接触式监测链路；以 Mosquitto 与 InfluxDB 为中心，构建面向监护场景的多组件协同时序数据处理架构；以 Grafana 为基础进行业务化适配重构，承载多设备管控与风险预警等核心业务能力。系统总体架构划分为设备感知层、数据传输层、时序数据中台层、可视化门户层四层，各层级间通过标准通信协议与接口实现互联互通，整体架构层次清晰、职责边界明确。

（1）设备感知层

由 LD6002 系列毫米波雷达与 ESP32-S3 边缘控制器构成，完成跌倒状态、人体存在状态及呼吸心率等生命体征数据的现场采集与前置处理，为全系统提供可靠原始数据源。

（2）数据传输层

基于 WiFi 无线链路 with MQTT 消息协议，建立边缘终端至云端的稳定传输通道，负责监测数据封装、异步上行及传输异常重试，保障多设备并发场景下数据传输的实时性与可靠性。

（3）时序数据中台层

以 Mosquitto 消息代理、InfluxDB 时序数据库为主体，配合数据处理脚本，统一实现设备消息接入、数据报文字段解析、时序数据入库存储、离线统计分析，并封装适配养老监护业务的数据查询、状态汇总与历史缓存回放能力，为上层门户提供标准化数据支撑。

（4）可视化门户层

基于 Grafana 完整 Web 架构进行养老监护场景业务化适配重构，在复用平台原生数据查询与状态分析能力的基础上完成场景定制配置；同时重构设计健康监

测专属门户，集成设备管理、卡片管理、风险弹窗告警、设备在线检测、分页筛选等配套管控功能，实现监护数据可视化展示、设备运维管理与异常风险预警的一体化承载。系统总体架构如图 3-1 所示。

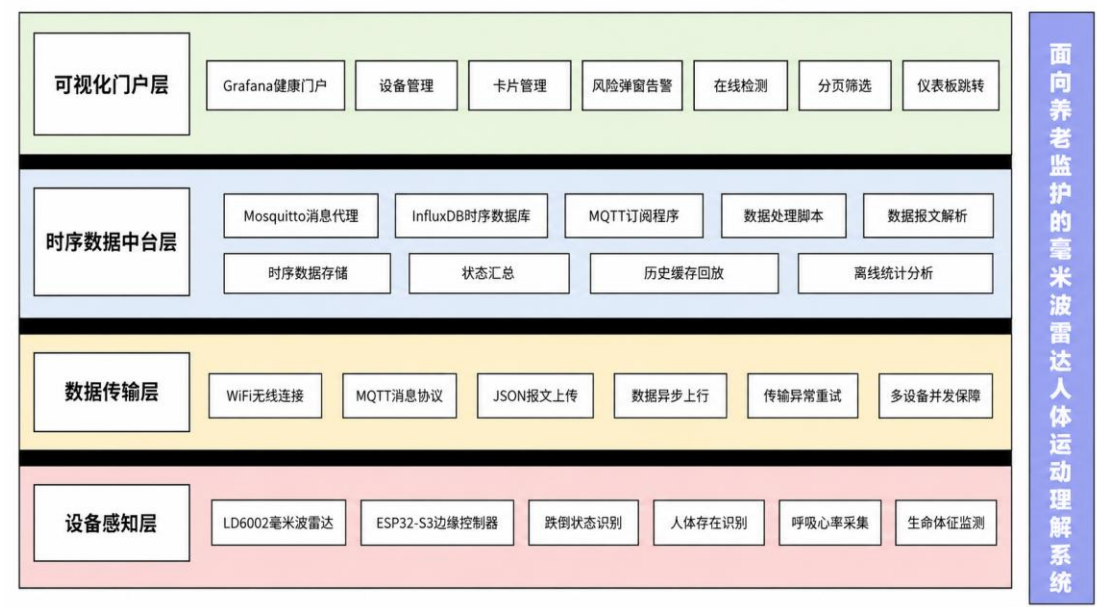


图 3-1 系统总体架构图

3.2. 系统数据流设计

系统数据流遵循“边缘生成、消息分发、结构化落库、缓存整合、门户展示、状态服务”的闭环路线。设备端采集状态后，以 JSON 形式通过 MQTT 上报；云端订阅程序收到消息后，将数据转换为字段并写入 InfluxDB；Grafana 后端扩展模块定时拉取实时数据，并将固定时段回放数据与实时数据合并写入缓存；前端业务门户通过/api/influxdb/query 获取缓存数据，而非每次直接查询数据库，从而降低前端查询压力并支持演示回放；服务层再基于同一数据源构建在线检测、风险提示与历史回放等监护能力。

在该闭环中，MQTT 主题承担了解耦设备端与平台端的作用，设备无需感知云端存储或展示逻辑；InfluxDB 则作为统一时序数据汇聚点，负责承接高频监测字段；Grafana 门户和监护服务进一步将底层数据转化为护理人员可直接理解的状态信息。由此，系统既保留了边缘采集的实时性，也兼顾了平台扩展与页面交互的灵活性。

3.3 功能模块设计

依据需求分析结论与系统总体架构，本系统在功能层面被划分为如下若干模块。模块之间通过 MQTT 主题、InfluxDB 测点与 HTTP 接口进行松耦合协作。系统功能模块划分如图 3-2 所示。

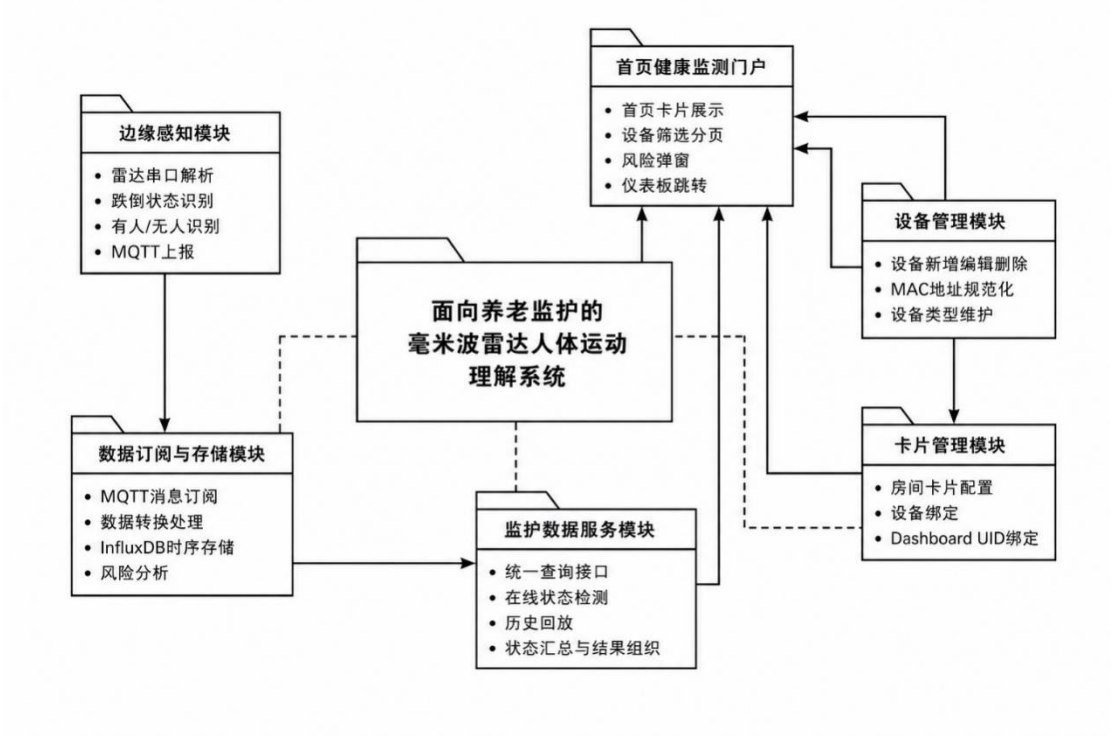


图 3-2 系统功能模块图

3.3.1 边缘感知模块

该模块负责接收雷达报文、解析跌倒与人体存在状态，并完成网络连接和 MQTT 发布。其输出字段至少包括 device_id、timestamp、fall、tx2 和 human。

3.3.2 数据接入与存储模块

该模块负责统一处理 MQTT 消息，并将解析后的结构化字段写入 InfluxDB 的 device_data 测量。对于十六进制报文，还需额外计算心率、呼吸、距离区间和体动幅度字段。

3.3.3 健康监测模块

该模块是系统功能完善过程中的重点内容。业务门户以卡片形式显示设备监测结果，针对不同类型设备给出差异化展示策略：心率设备显示心率、呼吸率、距离与体动趋势；血氧设备显示心率、血氧与最后有效帧时间；跌倒检测设备显示当前跌倒状态、持续时长与人体在位状态。

3.3.4 设备管理模块

设备管理模块提供设备实体的统一管理能力，用于维护设备名称、MAC 地址、设备类型和备注信息。设备表为后续门户配置和在线状态统计提供了基础数据。设备与卡片之间的配置关系如图 3-3 所示。

3.3.5 卡片管理模块

卡片管理模块管理门户展示规则。系统将每个门户卡片视为一个配置实体，记录房间名称、绑定设备、设备类型、绑定的仪表板 UID 等字段，实现业务门户展示的灵活调整能力。

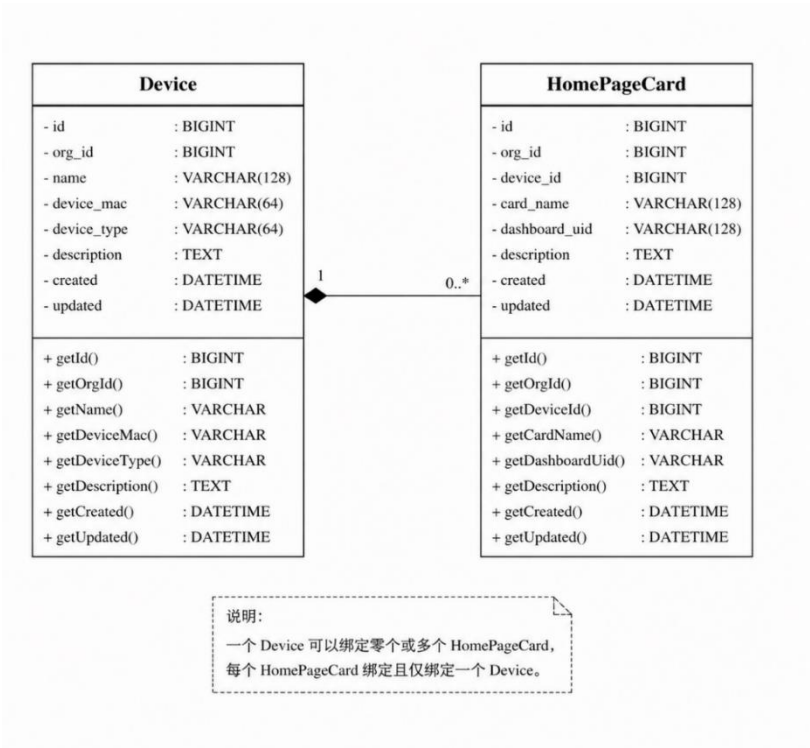


图 3-3 设备与卡片配置类图

3.3.6 监护数据服务模块

监护数据服务模块位于感知层与展示层之间，负责把原始时序数据转化为门户页面和业务模块可直接使用的结构化结果。模块核心能力包括生命体征查询、跌倒状态分析、设备在线检测、历史回放结果组织和统一接口返回，既降低了前端直接操作数据库的复杂度，也为风险提示、图表联动和多设备展示提供了稳定支撑。

3.4 数据模型设计

本系统包含关系型配置数据和时序监测数据两类模型。

3.4.1 设备配置模型

设备配置模型用于保存设备静态信息，包括设备名称、设备 MAC、设备类型、描述、创建时间和更新时间。系统在数据库中为 `org_id + device_mac` 建立唯一索引，确保同一组织内设备标识不重复。

3.4.2 门户卡片配置模型

门户卡片配置模型保存门户展示规则，包括绑定设备 ID、设备 MAC、设备类型、卡片名称与仪表板 UID。这样前端在渲染业务门户时即可通过卡片配置映射到实际设备和目标 Dashboard。

3.4.3 时序数据模型

时序数据统一保存在 InfluxDB 的 `device_data` 测量中，常见字段包括：

- `heart_rate_bpm`
- `respiration_bpm`
- `distance_min_cm`
- `movement_amplitude`
- `fall`
- `fall_count`

- spo2
- heart_rate_valid

关系型与时序型混合的数据模型使系统同时具备配置灵活性和监测性能。

值得注意的是，系统通过标准化的设备 MAC 将关系型配置数据与时序监测数据关联起来。设备表和门户卡片表使用统一的 MAC 规范化规则，时序库中的 device_id 也采用相同标识，因此设备管理、在线统计、卡片展示与风险分析能够围绕同一主键进行协同，降低了跨库关联的复杂度。

3.5 接口设计

系统对外接口按照 RESTful 风格组织，主要包括设备管理、卡片管理、时序数据查询代理与监护数据服务四类。所有接口均要求请求方处于登录状态，并以会话上下文中的 OrgID 作为隔离条件，而非直接信任前端传入的组织信息。

3.5.1 设备管理接口

设备管理接口包括：

- GET /api/devices: 获取设备列表；
- POST /api/devices: 新增设备；
- GET /api/devices/:id: 获取设备详情；
- PUT /api/devices/:id: 更新设备；
- DELETE /api/devices/:id: 删除设备。

3.5.2 门户卡片接口

门户卡片接口包括：

- GET /api/home-page-cards: 获取卡片配置列表；
- POST /api/home-page-cards: 新增卡片；
- PUT /api/home-page-cards/:id: 更新卡片；
- DELETE /api/home-page-cards/:id: 删除卡片。

3.5.3 InfluxDB 查询代理接口

系统提供 POST /api/influxdb/query 接口，前端提交 Flux 查询语句后，后端并不直接把查询发送到数据库，而是返回由后台缓存线程维护的最新结果。这种设计兼顾了实时展示、演示回放和查询负载控制。

结合接口实现可以看到，设备管理与卡片管理接口均要求用户处于登录状态，并以当前组织上下文中的 OrgID 作为隔离条件，而不是直接信任前端传入的组织信息。与此同时，卡片保存接口还会根据设备 ID 自动补齐设备 MAC 与设备类型，减少配置错误在前后端之间传播的风险。

3.6 监护数据服务设计

系统将监护数据服务定位为“数据组织层”。在总体设计中，后端围绕时序数据查询、设备状态汇总、缓存回放和结果封装四类职责构建统一接口，使业务门户、设备管理、测试模块以及智能体调用链路可以共享同一套数据访问与状态计算逻辑。该设计避免了页面端、问答调用端分别重复拼装查询语句的问题，也为后续扩展更多监护指标、风险判定规则和解释型服务能力预留了统一的数据组织入口。

从实现约束看，监护数据服务层并不直接暴露底层数据库细节，而是先对设备标识、时间窗口和指标范围进行统一处理，再向上提供结构化结果。这样做不仅可以避免前端页面重复拼装 Flux 查询语句，也能把设备在线检测、风险标记、历史回放等公共规则固化在服务侧；当智能体发起监测问答时，同样通过该服务层完成指标查询、结果组织与字段解释，从而保证业务门户展示结果与问答结果来源一致。对应到工程工作量上，该部分不仅包含接口封装，还涉及查询参数规范化、设备映射处理、缓存结果复用、风险状态组织以及面向智能体函数调用的结果适配，属于连接时序数据平台、业务门户和智能交互能力的关键中间层。监护数据服务与智能体之间的协同时序关系如图 3-4 所示。

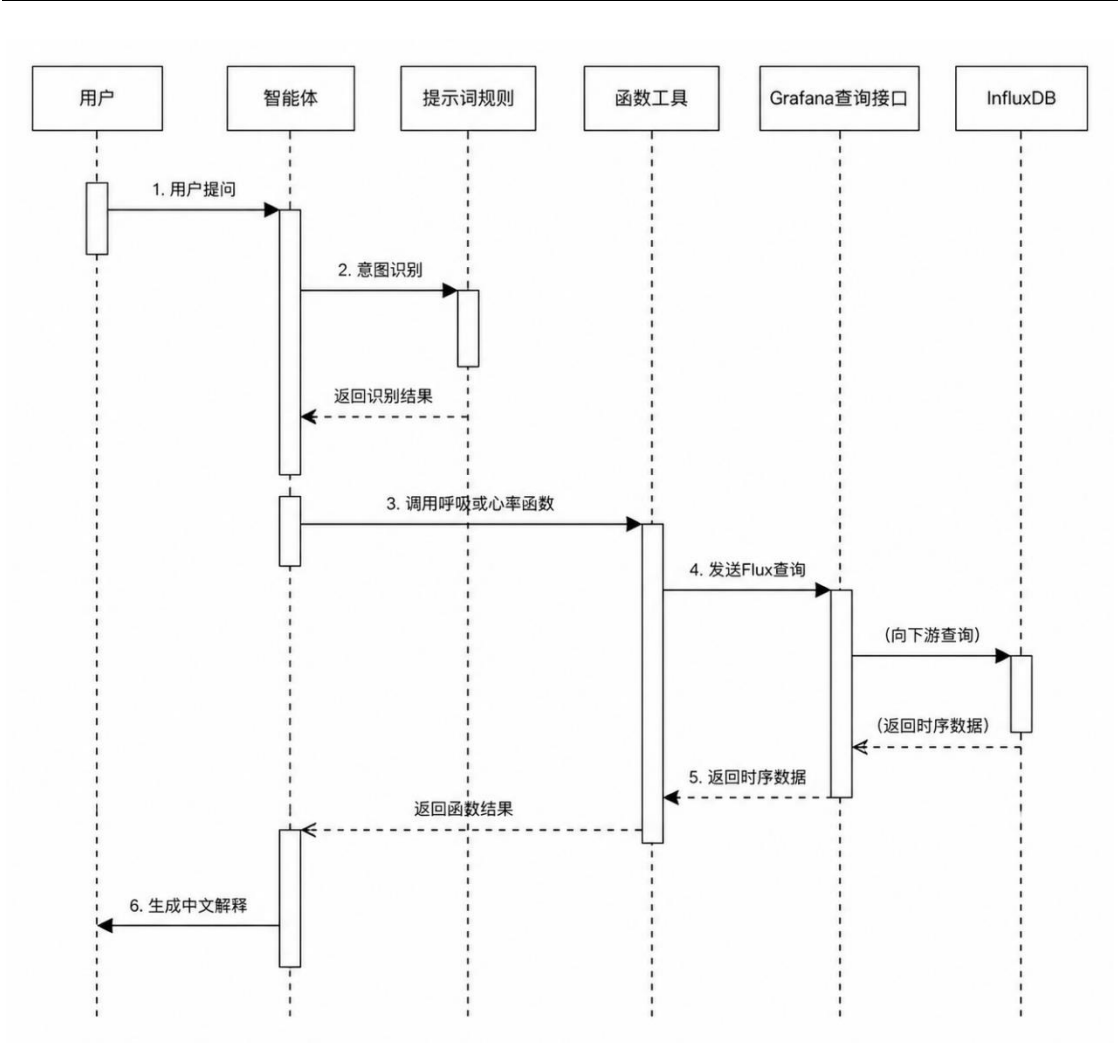


图 3-4 监护数据服务与智能体协同时序图

3.7 本章小结

本章从总体架构、数据流、功能模块、数据模型、接口设计与监护数据服务六个角度，对面向养老监护的毫米波雷达人体运动理解系统进行了总体设计。系统集边缘感知、时序平台、业务化门户与监护数据服务于一体，为第四章的详细实现提供了完整的设计依据。

第四章 系统详细设计与实现

本章在第三章总体设计的基础上，按照三条研究主线展开详细设计与实现，包括基于毫米波雷达与边缘控制器的感知与传输链路的设计与实现；面向监护场景的多组件协同时序数据处理架构的设计与实现；基于 Grafana 的养老监护业务化适配重构的设计与实现。

4.1 基于毫米波雷达的感知与传输链路实现

需要说明的是，本章在设备端报文解析、状态上报、风险联动和关键代码示例等环节中，主要选取跌倒检测设备作为贯穿示例。这是因为该类设备同时涉及跌倒状态、人体在位状态、告警提示与门户联动等核心业务流程，更便于完整呈现系统从边缘采集到页面反馈的实现链路。系统同时支持心率呼吸设备与血氧检测设备，相应的字段映射、卡片渲染与有效值处理将在 4.3 节业务化适配重构部分分别说明。本章选择统一示例仅为叙述清晰，不代表系统功能范围的限制。

4.1.1 硬件组成与通信方式

设备端采用 LD6002 系列毫米波雷达与 ESP32-S3 组成。ESP32-S3 一方面通过 UART 接收雷达数据帧，另一方面通过 GPIO 读取人体监测输出引脚状态。系统同时监听串口协议与 TX2 电平，形成双通道人体状态感知方式。

下图展示了本文系统最终形成的设备端产品形态。该设备以毫米波雷达模组和边缘控制器为核心，集成人体状态采集、串口通信边缘处理、无线联网和 MQTT 上报等功能，是系统感知层在实际部署中的主要硬件载体。

4.1.2 串口报文解析

串口解析逻辑围绕跌倒检测帧与人体存在状态帧展开，其中跌倒状态变化会触发 `last_fall_time`、`fall_count` 等变量更新，人体状态变化则用于记录有人或无人切换时刻。系统还额外记录 `total_frames`、`total_bytes` 和 `error_frames` 等串口统计量，用于反映设备在连续运行过程中的接收稳定性。

在设备固件当中，系统重点对两类报文进行处理，分别为跌倒检测帧以及人体存在状态帧。其中，跌倒检测帧主要用于更新 `current_fall_status` 与 `fall_count`；人体存在状态帧则用于判断区域内是否存在人员活动。相关实现位于主程序的解析函数当中，当识别到特定类型的字节序列时，系统就会读取校验字节以及状态字节，并对对应的状态变量进行更新。

为保证后续系统分析能力，固件在 MQTT 发布数据前对跌倒持续时间进行实时计算，并将其作为 `fall_count` 字段发送到云端。这一设计使业务门户能够显示“跌倒持续时长”，而不是仅给出单一布尔状态。

在内部状态组织上，固件分别维护 `fall_stats`、`human_stats` 和 `uart_stats` 等运行状态变量，用于记录当前跌倒状态、最后一次状态变化时间、累计串口帧数以及错误帧数量。通过这些状态量，系统不仅能够输出当前结果，还能够对跌倒持续时间、串口接收质量和运行稳定性进行持续监测，为后续链路排障提供依据。

4.1.3 WiFi 连接与 SoftAP 配网

设备上电后优先尝试连接 NVS 中保存的 WiFi 参数；若无历史配置，则使用默认 SSID 和密码；若连接失败，则切换到 SoftAP 配网模式，启动内置 HTTP 服务和 DNS 劫持，为手机端提供配网页面。该流程提高了设备部署和重配置的可操作性。

当前的固件端逻辑实现了对配网的健壮支持，设备端配网并非简单的热点开启，而是包含本地参数保存、网页表单提交、DNS 劫持引导和配置成功后自动重启等完整流程。该机制降低了养老房间现场部署门槛，避免每次更换网络环境都必须重新烧录程序。

从嵌入式实现角度看，系统利用事件组管理 WiFi 连接成功、连接失败和 MQTT 在线等关键状态，并在连接超时或重试失败后自动切换到配网模式。设备在保存新的无线网络参数后重新尝试连接，只有在网络恢复后才启动 MQTT 上报任务，从而保证链路建立顺序清晰且运行状态可控。

4.1.4MQTT 数据上报

系统以 1 秒为周期向 MQTT 主题发布状态消息。消息体中包含设备编号、毫秒时间戳、跌倒状态、GPIO 状态、跌倒持续秒数和人体存在状态。为保持本章叙述一致，以下以上报链路较完整的跌倒检测设备作为统一示例：

```
{  
  
  "device_id": "84F7035346E0",  
  
  "timestamp": 1745222400000,  
  
  "fall": 0,  
  
  "tx2": 0,  
  
  "fall_count": 0,  
  
  "human": 1  
}
```

该字段设计既满足门户展示，也满足后端告警与监护服务调用的需要。

在上报策略上，设备端仅上传门户展示与状态判断所必需的关键字段，而不把所有底层串口字节流直接转发到云端。这样既降低了消息体长度，又使后端处理逻辑能够围绕统一 JSON 结构展开，有利于提升消息解析效率和平台整体稳定性。设备端从状态解析到 MQTT 上报的工作流程如图 4-1 所示。

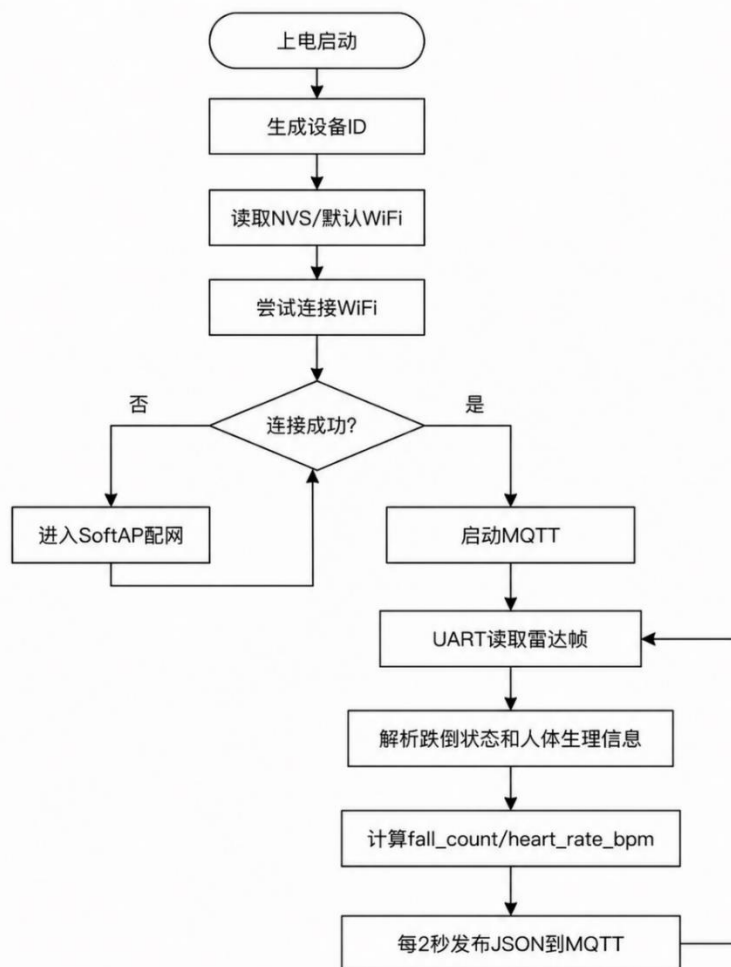


图 4-1 设备端工作流程

4.2 面向监护场景的多组件协同时序数据处理架构实现

除直接订阅程序外，系统还在主控制脚本中增加了定时查询线程，用于周期性执行风险检测逻辑。这样，消息接入与风险评估被拆分为“实时写入”和“周期分析”两条链路，既避免阻塞主订阅流程，又使平台能够持续输出面向监护场景的风险提示结果。

4.2.1 订阅流程

Mosquitto 作为系统的消息接入入口，承担多设备并发数据接入职责。云端使用 Python 程序调用 `mosquitto_sub` 持续订阅 `ld6002h/data/#` 主题，并将标准输出逐行送入数据处理器。处理器首先尝试将负载解析为生命体征原始帧；若失败，

再按 JSON 方式解析。此设计兼容两类上报格式，使平台既能接收生命体征原始报文，也能接收跌倒和人体存在状态的 JSON 消息。

生命体征原始帧的解析过程具有较明确的格式约束：数据包以 AA55 为帧头，以 0D0A 为结束标记，长度固定为 15 字节。解析完成后，系统可进一步得到 occupied、respiration_bpm、heart_rate_bpm、distance_min_cm、distance_max_cm、movement 和 movement_amplitude 等字段，并将保留字节转写为十六进制字符串，便于后续排查原始报文问题。云端消息接入与入库流程如图 4-2 所示。

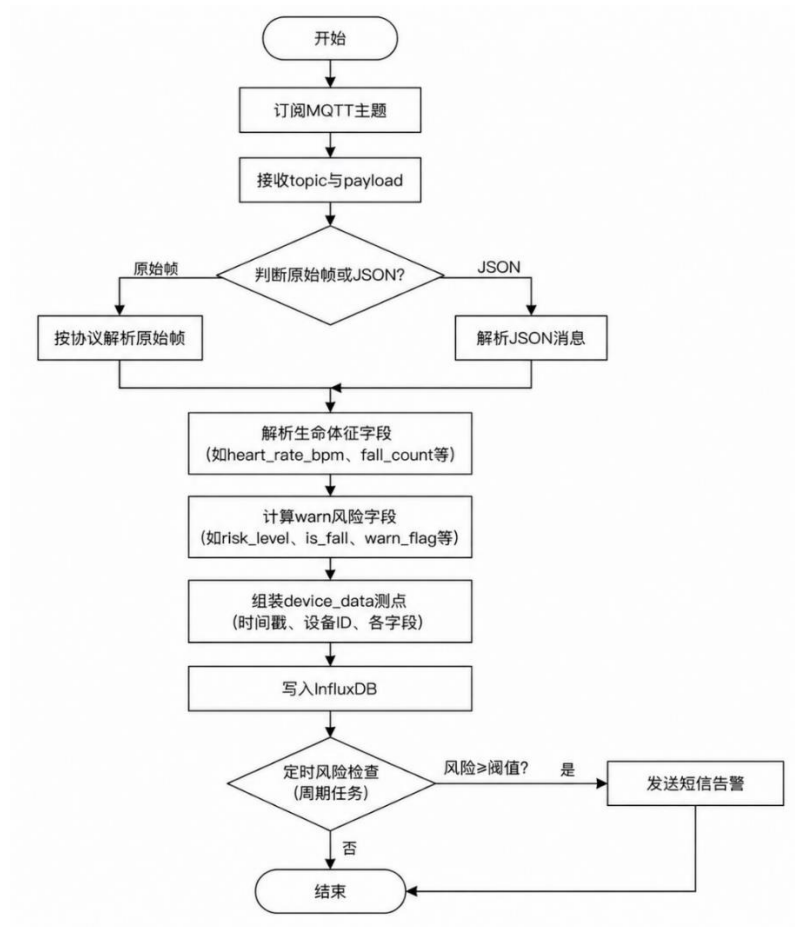


图 4-2 云端消息接入与入库流程图

4.2.2 时序字段映射

字段解析承担将原始消息转化为结构化字段的职责，针对两类报文分别实现解析逻辑。在数据入库阶段，系统统一使用 device_data 作为测量名称，并将

device_id 与 topic 作为标签，其他整型、浮点型或布尔型数据作为字段写入 InfluxDB。这样既便于按设备检索，也便于按字段聚合。

这种“标签 + 字段”的入库方式与 InfluxDB 的时序建模特点相匹配。设备标识与主题名称作为标签，便于按设备快速过滤；心率、呼吸、跌倒状态、体动幅度等作为字段，便于执行区间统计、均值计算和时间窗口分析。此外，系统还保留了原始 raw_data 字段，用于在必要时回溯上报原文，提高排障与复核效率。

对于生命体征原始帧，系统会解码出：

是否有人；呼吸频率；心率；距离区间；是否体动；体动幅度等

在告警辅助字段设计上，系统并未直接把阈值判断结果写死到页面逻辑中，而是在入库阶段额外生成 warn 字段。当前规则主要围绕心率和呼吸频率是否落在合理区间展开，例如心率偏离 60~120 次/分或呼吸频率偏离 10~20 次/分时标记异常。这样既便于业务门户进行颜色提示，也方便后续在风险脚本中继续组合分析。对应的 InfluxDB 面板示例如图 4-3 所示。

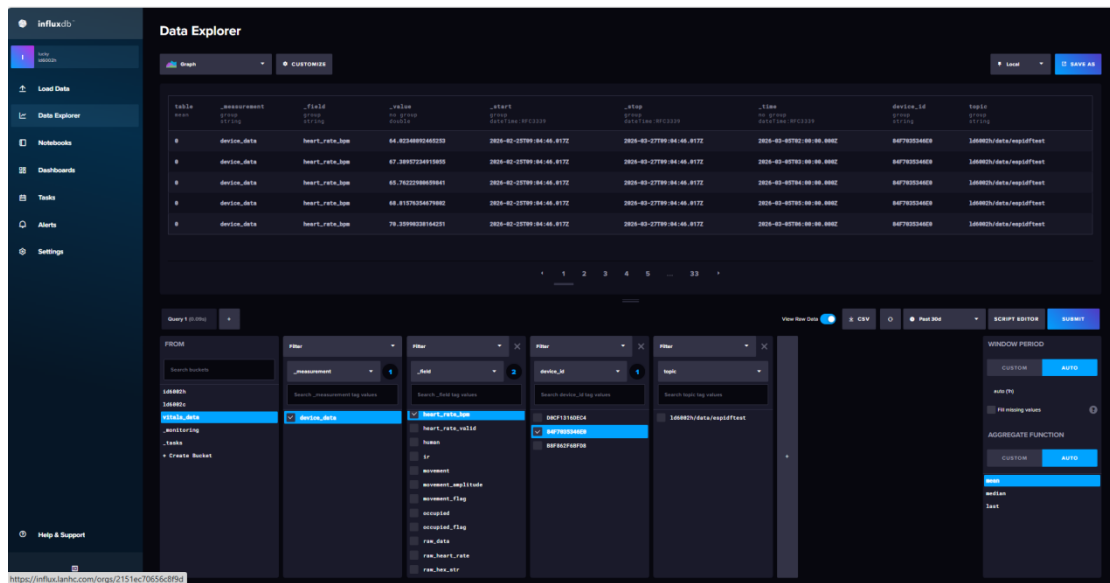


图 4-3 InfluxDB 面板示例

4.2.3 风险检测与短信告警

系统设计了基于 InfluxDB 查询的风险检测脚本，用于计算近时间窗内的体动、心率和呼吸组合特征，并识别可能的跌倒风险事件。当触发条件满足时，系

统会调用短信接口发送提醒。出于安全与隐私考虑，本文不展示具体短信接口凭据，但该功能表明系统已具备从传感器感知到风险告警外发的闭环能力。

在风险判定实现中，系统先按设备编号从 InfluxDB 查询最近一分钟内的 movement_amplitude、heart_rate_bpm 和 respiration_bpm 字段，再按时间戳将同一时刻的多字段记录聚合为一帧。随后，系统以体动幅度大于 900 且呼吸频率大于 20 或心率大于 100 作为风险触发条件，并将满足条件的记录加入风险事件列表。风险告警短信发送结果如图 4-4 所示。

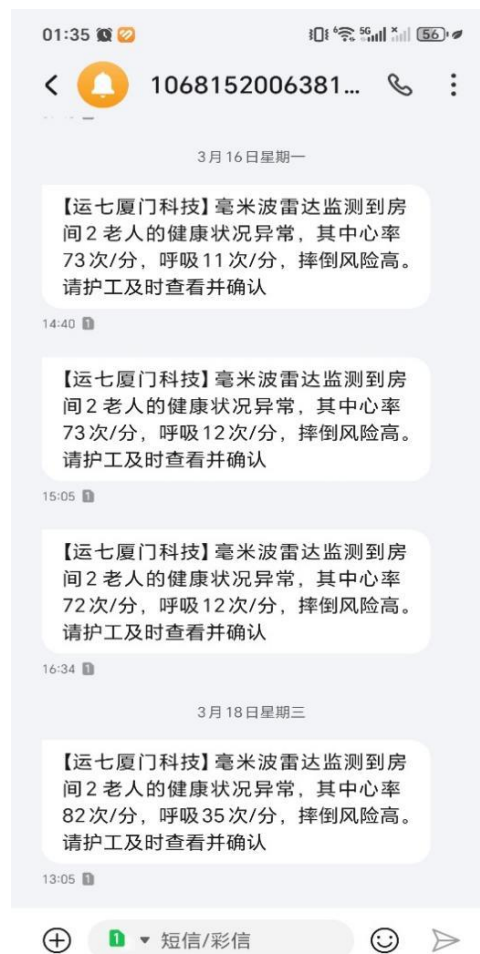


图 4-4 风险告警短信发送

4.3 基于 Grafana 的养老监护业务化适配重构

本文在 Grafana 框架基础上开展面向养老监护场景的业务化适配重构。重构工作利用 Grafana 插件机制、前端可扩展能力与后端服务扩展接口，定制构建一系列面向监护场景的业务模块。具体包括设备配置列表、实时生命体征结果、在

线设备集合、分页状态、设备筛选状态、风险弹窗状态、设备管理模态框状态等。其中，设备过滤与分页逻辑通过记忆化计算得到，风险状态则通过对跌倒类卡片结果进行实时扫描触发。

4.3.1 业务门户重构与状态组织

门户的前端核心实现位于 `HomePage.tsx` 中。前端通过 `React` 的 `useState` 与 `useEffect` 管理以下核心状态：

- (1) 卡片配置列表；
- (2) 当前设备生命体征列表；
- (3) 设备实体列表；
- (4) 在线设备集合；
- (5) 当前分页与过滤条件；
- (6) 风险弹窗与已确认房间集合；
- (7) 设备管理与卡片管理模态框状态。

该实现说明业务门户已从传统的静态欢迎页升级为具备配置、查询、展示、交互和告警功能的业务门户。

4.3.2 多类型卡片渲染策略

系统对设备类型进行了抽象，支持 `heart-rate`、`blood-oxygen` 和 `fall-detection` 三类设备：

对于心率类设备，业务门户显示心率、呼吸率、距离和体动幅度；

对于血氧类设备，业务门户显示心率、血氧和最后有效帧时间；

对于跌倒类设备，业务门户显示跌倒状态、跌倒持续时间和人体在位状态。

当前项目中业务门户对设备类型采用 `heart-rate`、`blood-oxygen` 和 `fall-detection` 三种标识进行区分，并在同一套数据获取流程下完成差异化字段解释、状态计算和界面渲染。也就是说，前端虽然共用统一的查询接口，但会根据 `deviceType` 选择不同的字段组合和显示策略。

对于心率呼吸设备，系统重点使用 `heart_rate_bpm`、`respiration_bpm`、`distance_min_cm` 和 `movement_amplitude` 四类指标。处理过程中，前端会为心率、

呼吸率、距离和体动值分别维护趋势方向与过期标记，并将 `heart-rate` 设备的 `fallRisk` 固定为 `false`，避免因体动值异常而直接套用跌倒告警样式；与此同时，设备在位状态主要依据心率数据是否有效进行判断，因此更适合承担日常生命体征观察任务。

对于跌倒检测设备，系统重点读取 `fall`、`fall_count` 和 `human` 三类字段。其中，`fall` 字段被转换为 `fallDetected` 状态，`fall_count` 被解释为跌倒持续时间，`human` 字段被进一步转换为有人或无人状态。前端对这类设备不再采用通用指标卡片，而是调用专门的 `renderFallDetectionCard` 渲染逻辑，将页面划分为跌倒状态、跌倒计时和人员监测三个区域；当跌倒持续时间超过设定阈值后，系统还会联动风险弹窗与确认冷却机制，从而形成更适合养老看护场景的风险提示界面。

除了基础的数据展示，前端还额外加入了趋势箭头、异常状态着色、卡片跳转以及陈旧数据灰显等交互反馈。例如跌倒风险卡片采用红色边框和阴影强调风险状态，在位状态则通过背景颜色辅助表达。

还有一个容易被忽视但严重影响用户体验的问题——数据刷新带来的页面抖动。养老监护场景里，心率、呼吸、血氧这些指标并不会每一秒都稳定返回有效值，如果前端一检测不到数据就立刻清空卡片，会导致卡片空白，看起来像系统故障。为此，页面内部专门保留了“最后一次有效值”以及“陈旧字段”两类状态，对系统的数据传输波动做出处理。也就是说，在某个采集周期内即便没有新的有效数据，系统也不会马上把数据显示区域置空，而是暂时沿用最近一次结果，同时通过灰显提示用户：这部分数据已经过期。

这样处理是为了给界面加了一层“缓冲”。数据显示不会忽明忽暗，但状态真实性又没有被掩盖，放在养老监护这种高频、小周期刷新的业务场景里，会显得更稳，也更符合实际使用习惯。

在数据组织过程中，系统还将心率、呼吸率、距离、体动值、血氧和跌倒状态分别封装为统一的渲染项，并结合趋势方向与过期标记进行界面展示。这样既便于多类型设备复用同一套卡片渲染框架，也降低了后续新增设备类型时的前端改造成本。

4.3.3 分页、筛选与手动刷新

设备数量增多时，业务门户支持分页与按设备类型筛选。如图 4-5，为了避免直接将大量卡片堆叠在单页中，系统设置了每页固定数量并结合按钮切换页码。该设计明显提升了多房间场景下的可读性。同时延用了 Gafana 原有的组织管理，登录后通过 orgID 区分不同组织用户访问的门户设备展示和仪表板权限。

4.3.4 跌倒风险弹窗机制

业务门户通过对 deviceVitals 中跌倒类卡片的状态进行监听，在跌倒持续时间超过阈值后弹出告警窗口，并维护确认冷却时间，避免重复提醒。该机制兼顾了风险及时性和交互友好性。同时可以点击业务门户查看对应设备一段时间内的详细数据折线图，如图 4-6 所示。

风险弹窗机制还引入了确认冷却时间和已确认房间集合，用于控制重复告警频率。当同一房间在短时间内持续处于风险状态时，系统不会无限次弹出相同提示，而是在保留风险标识的同时降低交互干扰，这更符合养老监护场景下的实际使用习惯。



图 4-5 业务门户监控界面示例

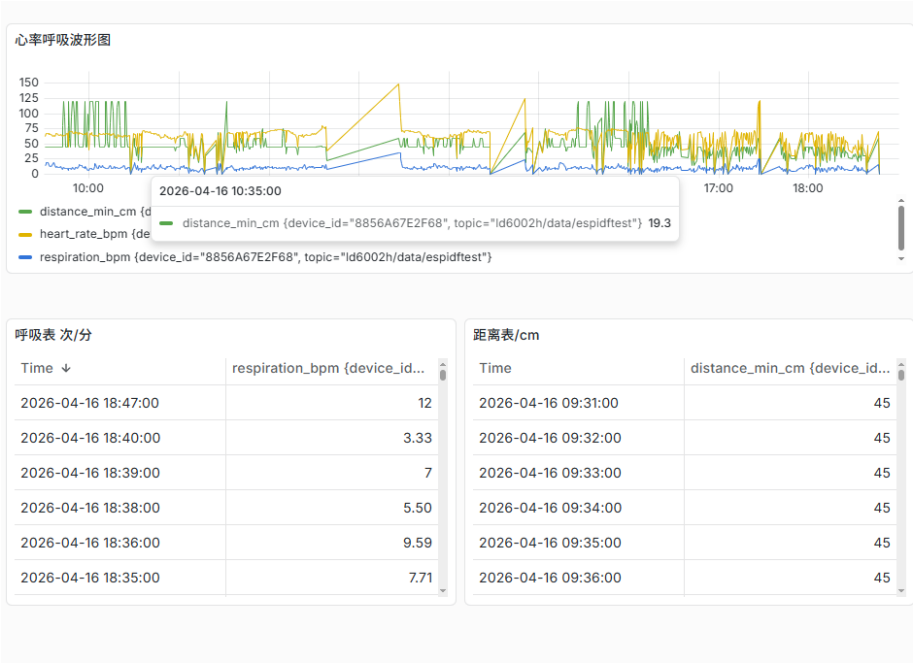


图 4-6 仪表板数据折线图界面

4.4 设备管理模块实现

4.4.1 设备数据模型

设备管理模块定义了 Device 结构体，包含 id、org_id、name、device_mac、device_type、description、created 和 updated 等字段。系统在写入前会对设备名称、MAC 地址和设备类型进行规范化处理，其中 MAC 地址统一去除分隔符并转为大写，以便与时序数据中的 device_id 对齐。

在服务层实现中，设备实体采用关系型结构进行持久化保存，字段覆盖名称、设备标识、设备类型、备注、创建时间和更新时间等内容。系统还通过标准化处理将 MAC 地址中的分隔符统一移除并转为大写，以保证其能与 InfluxDB 中的 device_id 直接对应。

4.4.2 设备管理 API

后端通过 Go 语言在 HTTPServer 中注册设备管理路由，实现增删改查能力。服务层负责命令校验，存储层负责数据库操作。新增唯一索引后，同一组织内的

设备 MAC 不允许重复，避免多个设备配置冲突。设备管理接口调用与存储交互时序如图 4-7 所示。

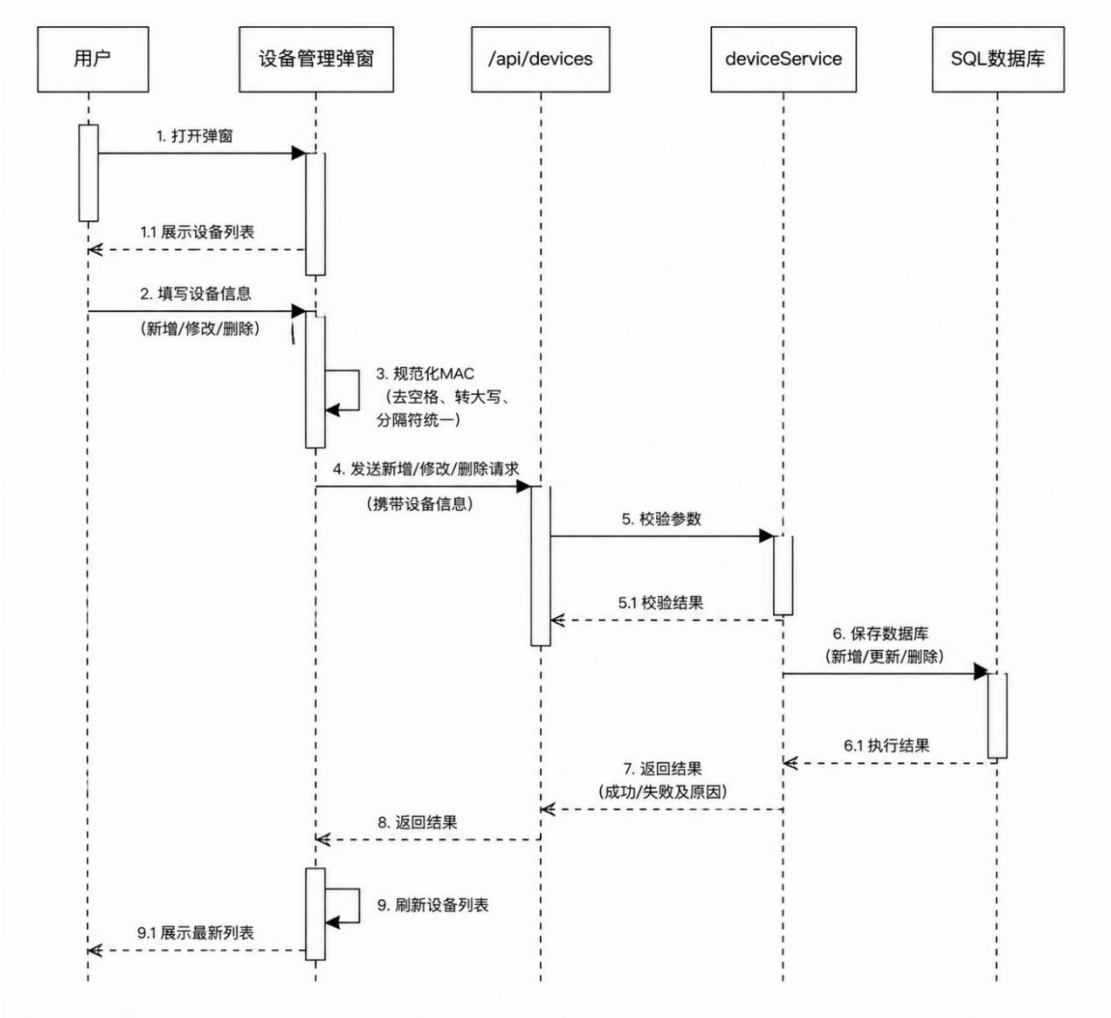


图 4-7 设备管理时序图

4.4.3 前端设备管理弹窗

在门户配置弹窗中，用户可进入设备管理窗口，对设备进行新增、编辑与删除操作。设备表单包含设备名称、设备 MAC、设备类型和备注字段，下方表格用于展示当前系统已录入的设备列表。该设计将原本依赖手工修改代码的维护方式转化为可视化管理操作。具体界面如下图 4-8 及图 4-9 所展示。

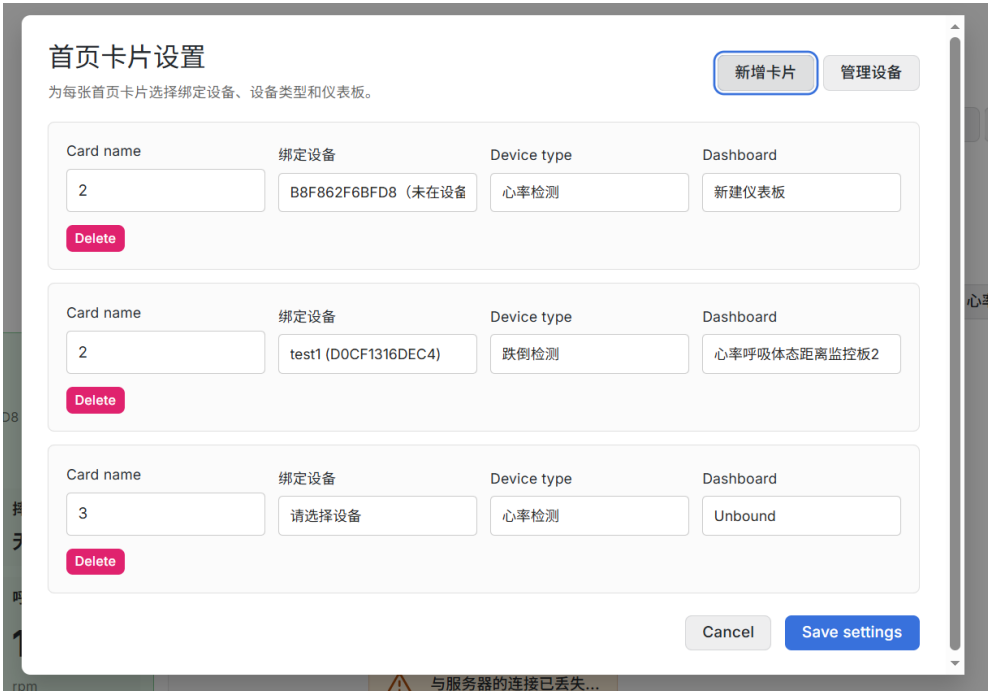


图 4-8 业务门户房间卡片管理示例



图 4-9 业务门户设备管理示例

4.5 卡片管理模块实现

4.5.1 卡片配置模型

卡片配置实体 HomePageCard 包含设备 ID、设备 MAC、设备类型、卡片名称和 Dashboard UID 等字段。系统在保存卡片前，会自动根据设备 ID 回填 device_mac 和 device_type，以减少前端输入错误。

卡片配置与设备实体并非简单的字符串拼接关系，而是通过设备 ID、设备 MAC 与设备类型进行多重映射。后端在保存卡片配置时会根据设备 ID 自动补齐设备标识信息，再将仪表板 UID 进一步转换为可访问的 Dashboard 地址，从而保证门户卡片、设备实体和仪表板跳转目标之间保持一致。

从前端保存逻辑看，卡片设置并不是全量删除后重建，而是根据是否已存在主键标识区分新增、更新和删除三类操作。该差量化处理方式能够减少不必要的数据库写入，也有利于在后续扩展审计与版本管理能力。

4.5.2 卡片管理逻辑

卡片设置窗口允许用户为每一行配置房间名称、绑定设备、设备类型与仪表板。系统在保存前执行以下校验：

- (1) 至少保留一个卡片；
- (2) 每个卡片必须绑定一个设备；
- (3) 同一设备只能绑定一张卡片；
- (4) 选中的仪表板 UID 应能映射到可用 Dashboard。

对于已删除的卡片，前端会调用删除接口；对于已有卡片执行更新；对于新增卡片执行新增。这样实现了差量化配置保存。

为防止配置关系失配，系统在保存卡片配置时还会检查目标仪表板是否可映射到有效 Dashboard 地址。这样可避免出现门户卡片已创建但点击后无法跳转的情况，从而提高门户配置能力的可用性。具体卡片保存时序图 4-10 所示。

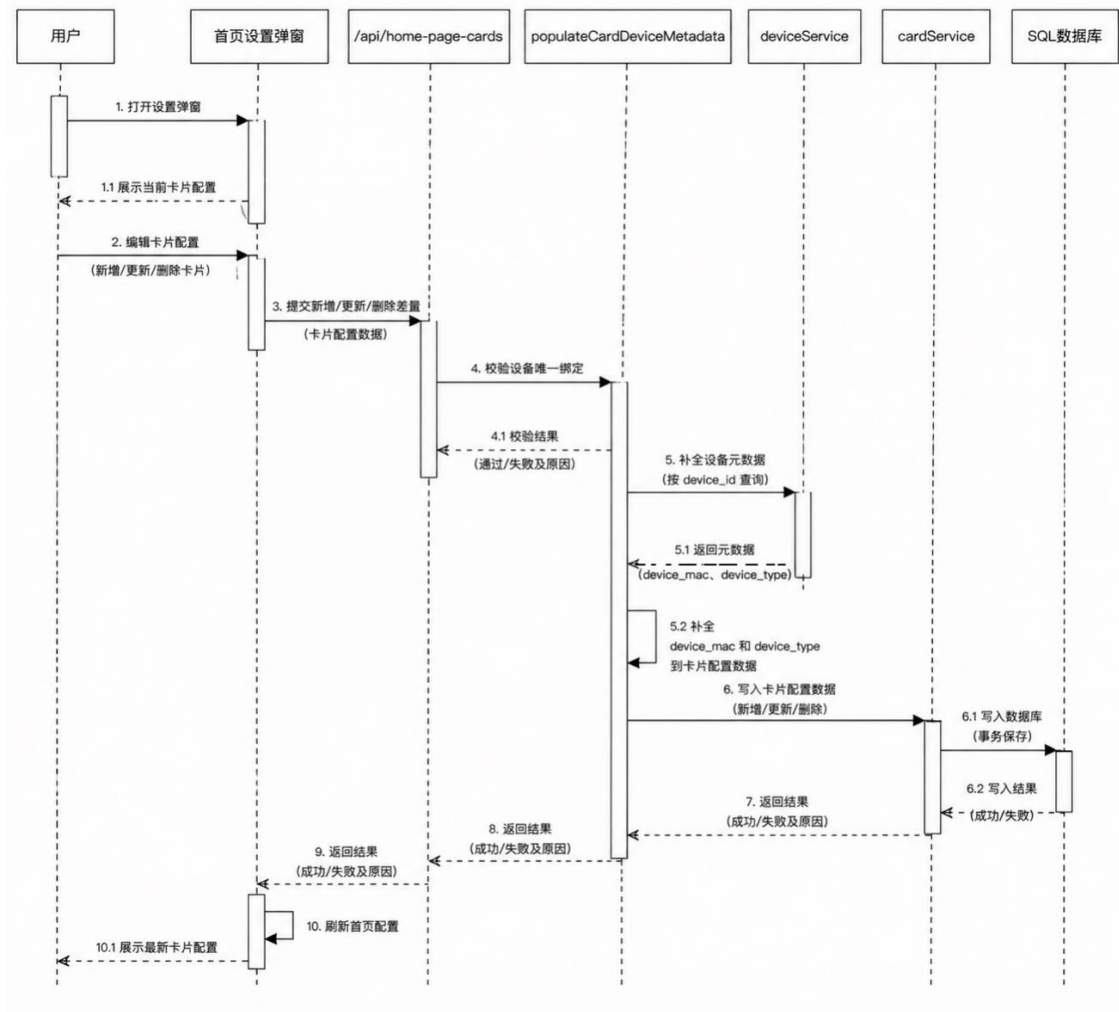


图 4-10 卡片管理时序图

卡片管理模块使业务门户具备配置化管理能力。相较于直接对房间和设备信息进行硬编码，配置化设计更适用于养老机构、家庭多房间和病区多床位等场景，也更符合软件系统在可维护性与可扩展性方面的要求。

4.6 InfluxDB 缓存代理与回放机制实现

4.6.1 查询代理设计

为了避免前端频繁直接访问 InfluxDB 带来的查询压力，系统在后端额外封装了 InfluxDBQuery 接口。这个接口并不会在每次请求到来时都重新执行完整的 Flux 查询，而是优先返回后台缓存线程提前维护好的最新结果。换句话说，真正

的数据拉取与整理工作，被集中放在服务端持续完成，浏览器端只负责获取已经处理后的数据内容。

这样处理之后，前端不再需要频繁组织 Flux 语句，也不必承担查询失败后的重试逻辑。尤其在设备数量逐渐增加时，如果大量浏览器同时直接访问时序数据库，查询波动会非常明显，甚至可能影响页面响应稳定性。增加这一层查询代理后，相当于在 InfluxDB 与业务门户之间加入了一道缓冲层，数据访问压力被统一吸收，系统整体运行会平稳很多。

从实际效果来看，这种方式不仅减少了数据库瞬时查询负载，也降低了前端页面对数据库状态的敏感程度。即使后端某一时刻出现短暂查询延迟，门户页面依然能够优先读取缓存中的最新结果，不至于立即出现页面空白或指标闪烁的问题。

4.6.2 固定时段回放与实时合并

系统后端启动后，会先从 InfluxDB 中读取固定时间窗口内的历史记录，并将其作为初始回放数据加载到缓存中。随后，后台线程再持续接收实时设备数据，并不断将新到达的数据与历史缓存进行合并更新。

这一机制的作用，其实更偏向“维持展示连续性”。在养老监护场景里，实时数据并不一定始终稳定，例如网络波动、设备短时离线或者 MQTT 消息延迟，都可能导致某个时间段内的数据暂时缺失。如果页面完全依赖实时流刷新，一旦数据中断，门户界面就容易出现大面积空白或者状态跳变。通过提前加载固定时间窗口的历史回放数据，即便实时数据暂时不足，页面也仍然能够维持基本展示状态。

具体实现时，系统还针对不同类型的数据采用了差异化处理方式。心率、呼吸频率、血氧等生命体征更强调实时变化，因此查询范围通常只覆盖最近几秒的数据；但在线状态检测如果也采用相同策略，就会变得过于敏感。一次短暂网络抖动、一次消息延迟，都可能让设备被误判为离线。所以系统会适当拉长在线检测的时间窗口，用更宽松的方式进行状态确认，以减少误判情况。

另外，历史回放数据与实时流数据并不是彼此独立渲染的，而是被统一整理进同一套缓存结构中。页面首次加载时，会优先利用历史回放数据补齐整体卡片内容，确保门户打开后能够立即显示完整监护信息；之后新增的实时记录，则只对关键指标进行增量更新。

某种程度上，这套设计更像是在“真实性”和“可读性”之间找平衡。系统既没有为了界面平滑去伪造实时状态，也没有因为追求绝对实时而让页面不停闪烁、抖动。对于需要长时间持续运行的养老监护门户来说，这种处理方式会更耐用，用户也更容易建立信任感。

4.7 监护数据服务实现

4.7.1 查询服务目录组织

系统在后端侧将监护数据服务划分为生命体征查询、跌倒状态分析和设备在线检测三类能力，分别面向门户展示、风险提示和设备管理场景提供数据支撑。

生命体征查询服务负责从 InfluxDB 中提取心率、呼吸率、距离和体动等关键字段，并结合时间窗口生成可用于图表展示和状态判断的结果集；

风险状态服务负责根据跌倒持续时间、人体在位状态和最新时序记录生成告警标记，同时为弹窗提示和卡片高亮提供统一依据。

这些服务通过统一接口向前端返回结构化 JSON 结果，从而降低业务页面与底层时序查询之间的耦合度。

在风险状态组织方面，后端会结合体动幅度、心率、呼吸、跌倒持续时间和人体在位状态等信息生成统一结果对象。这样一来，门户卡片高亮、风险弹窗和后续历史追溯都可以围绕相同的数据结构展开，避免不同页面各自维护一套判断逻辑。

4.7.2 时间窗口与设备映射处理

系统查询接口支持时间戳、绝对日期和固定时段范围等多种时间表达，并在服务内部统一换算为数据库可识别的查询窗口。这种设计既满足业务门户实时刷新，也适应演示回放和历史追溯等场景。

在设备映射方面，系统根据设备表与门户卡片配置表建立“房间—设备—仪表板”的对应关系，使页面展示、在线统计与风险判断能够围绕同一设备标识展开，避免出现配置项与监测数据脱节的问题。

系统在组织查询参数时，并不会把所有请求都按同一种方式处理，而是根据使用场景进行区分。实时展示更强调时效性，因此只会关注最近几秒内产生的数据；在线状态检测则相对“保守”一些，会采用更长的时间窗口进行判断，避免因短时网络波动或消息延迟造成设备误离线

这种分层式的数据查询策略，本质上是在不同需求之间做取舍。实时页面需要刷新足够快，状态反馈不能拖延；而养老监护场景又要求数据尽量连续，不能频繁出现空档或状态跳变。将三类场景拆开处理后，系统既保留了实时响应能力，也让整体监护过程看起来更加稳定、连贯。

4.7.3 查询参数与结果组织规则

为保证页面调用稳定性，服务层对查询参数进行了统一约束，包括时间范围、设备编号、指标名称和分页条件等。接口返回值中同时保留原始字段、状态标记、更新时间和趋势说明，便于前端在卡片、弹窗和仪表板跳转中复用相同数据结构。

在实现策略上，服务层将设备编号、时间范围、指标名称和房间映射关系统一封装后再返回结果。这样不仅能够减少前端对底层查询细节的感知，也便于在后续增加新的监护指标时复用既有接口结构。

从工程价值看，监护数据服务层并不直接替代底层监测功能，而是将采集、存储、分析与展示环节串联起来。其工程价值在于把分散的时序记录转化为面向养老监护场景的业务结果，提升了系统的可维护性、可扩展性和页面开发效率，也为后续接入更多传感器和监护规则预留了接口空间。进一步看，监护数据服务层还承担了“统一数据出口”的作用。无论前端页面调用、风险判断脚本调用，还是后续可能增加的移动端或护理端接口调用，都可以围绕统一的数据结构展开，从而减少重复开发工作，并提高不同业务模块之间的一致性。

4.8 关键实现代码说明

为避免正文中堆积过长源码，本文仅选取设备端上报、门户卡片绑定和设备在线检查三类代表性逻辑进行说明。这三类代码分别对应“边缘端生成统一数据结构”“业务门户配置设备与仪表板关系”“前端按最近时间窗口判断设备在线状态”三个关键环节，能够反映系统从底层采集到上层业务展示的主要实现路径。

4.8.1 设备端上报结构

设备端以 JSON 作为统一上报格式，将跌倒状态与人体存在状态合并发布，确保门户卡片展示、风险判断与函数查询使用同一份输入数据。

在本项目中，数据由 ESP32 每秒获取并上报，字段包含设备标识、时间戳、跌倒状态、TX2 引脚电平、跌倒持续时间与人体存在状态。对应的 MQTT 发布任务实现如图 4-11 所示。

```

287 static void mqtt_publish_task(void *arg)
288 {
289     ESP_LOGI(TAG, "✅ MQTT发布任务已启动 (每1s发送一次)");
290
291     while (1) {
292         vTaskDelay(pdMS_TO_TICKS(1000));
293
294         if (s_provisioning_mode) continue;
295         if (!s_mqtt_client || !s_mqtt_event_group) continue;
296
297         EventBits_t bits = xEventGroupGetBits(s_mqtt_event_group);
298         if ((bits & MQTT_CONNECTED_BIT) == 0) { ...
299
300         int tx2_level = gpio_get_level(TX2_GPIO);
301         int64_t timestamp_ms = esp_timer_get_time() / 1000;
302
303         uint32_t fall_duration = 0;
304         if (fall_stats.current_fall_status == FALL_STATUS_DETECTED &&
305             fall_stats.last_fall_time != 0) { ...
306         fall_stats.fall_count = fall_duration;
307
308         char payload[256];
309         int written = snprintf(payload, sizeof(payload),
310             "{\"device_id\":\"%s\",\"timestamp\":%lld,\"
311             \"fall\":%d,\"tx2\":%d,\"fall_count\":%\"PRIu32\", \"
312             \"human\":%d\",
313             s_device_id,
314             (long long)timestamp_ms,
315             (int)fall_stats.current_fall_status,
316             tx2_level,
317             fall_duration,
318             (int)human_stats.current_human_status);
319
320         if (written <= 0 || written >= (int)sizeof(payload)) { ...
321
322         int msg_id = esp_mqtt_client_publish(s_mqtt_client, MQTT_TOPIC, payload, 0, 1, 0);
323         if (msg_id >= 0) { ...
324         } else { ...
325     }
326 }

```

图 4-11 跌倒设备 mqtt 发布任务

该代码片段中的 `device_id` 用于标识设备，`timestamp` 用于标识上报时间，`fall` 和 `human` 分别表示跌倒状态与人体存在状态，`tx2` 保留雷达状态引脚信息，`fall_count` 记录跌倒持续时间。通过在设备端统一字段名称，云端订阅程序、InfluxDB 入库逻辑以及 Grafana 业务门户可以围绕同一数据结构展开处理，避免不同模块之间重复转换字段语义。

4.8.2 门户卡片绑定逻辑

门户卡片绑定逻辑位于 `HomePage.tsx` 和 `home_page_card.go`。前端打开设置弹窗后调用 `/api/home-page-cards` 读取当前配置，将 `cardName`、`deviceId`、`deviceMac`、`deviceType`、`dashboardUid` 等字段转换为页面中的卡片草稿；保存时先清洗空白项并检查设备绑定是否重复，再根据原有卡片 ID 执行新增、更新或删除操作。前端卡片配置处理逻辑如图 4-12 所示。

```

1082  const saveSettings = async () => {
1083    const cleanedRows: DeviceConfig[] = settingsDraft.map((item): DeviceConfig => ({
1084      ...item,
1085      room: item.room.trim(),
1086      deviceId: item.deviceId ?? null,
1087      deviceMac: normalizeDeviceMac(item.deviceMac ?? ''),
1088      deviceType: (item.deviceType ?? DEFAULT_DEVICE_TYPE).trim() || DEFAULT_DEVICE_TYPE,
1089      dashboardUid: (item.dashboardUid ?? '').trim(),
1090      dashboardUrl: (item.dashboardUrl ?? '').trim(),
1091    }));
1092
1093    if (cleanedRows.length === 0) {
1094      setSettingsError('Please keep at least one card.');
```

```

1095    return;
1096  }
1097
1098  if (cleanedRows.some((item: DeviceConfig) => !item.room || item.deviceId == null)) {
1099    setSettingsError('请选择卡片绑定的设备。');
1100    return;
1101  }
1102
1103  const uniqueDeviceIds = new Set(cleanedRows.map((item: DeviceConfig) => item.deviceId));
1104  if (uniqueDeviceIds.size !== cleanedRows.length) {
1105    setSettingsError('每个设备只能绑定一张卡片。');
1106    return;
1107  }
1108
1109  setIsSavingSettings(true);
1110  setSettingsError(null);
1111
1112  try {
1113    const rowsWithId = cleanedRows.filter(...
1114  );
1115    const nextIds = new Set<number>(rowsWithId.map((item: DeviceConfigWithId) => item.id));
1116
1117    const deletedIds = deviceConfigs
1118      .filter((item: DeviceConfig): item is DeviceConfigWithId => typeof item.id === 'number')
1119      .filter((item: DeviceConfigWithId) => !nextIds.has(item.id))
1120      .map((item: DeviceConfigWithId) => item.id);
1121
1122    for (const id of deletedIds) {
1123      await getBackendSrv().delete(`/api/home-page-cards/${id}`);
1124    }
1125  }
1126
```

图 4-12 卡片管理前端具体实现

后端接口为卡片配置提供持久化入口，相关路由注册如图 4-13 所示。支持列表查询、新建、更新和删除。创建或更新时，后端会根据 deviceId 补齐设备 MAC 和设备类型，并使用当前登录上下文中的 OrgID 作为组织边界，避免前端直接传入组织信息造成配置越权。该设计使“房间名称—设备 MAC—设备类型—Dashboard UID”形成稳定映射，前端卡片展示、仪表板跳转和设备在线统计都可以复用同一份配置。

```

16  )
17
18  func (hs *HTTPServer) registerHomePageCardAPI(apiRoute routing.RouteRegister) {
19      reqSignedIn := middleware.ReqSignedIn
20
21      apiRoute.Get("/home-page-cards", reqSignedIn, routing.Wrap(hs.listHomePageCards))
22      apiRoute.Post("/home-page-cards", reqSignedIn, routing.Wrap(hs.createHomePageCard))
23      apiRoute.Put("/home-page-cards/:id", reqSignedIn, routing.Wrap(hs.updateHomePageCard))
24      apiRoute.Delete("/home-page-cards/:id", reqSignedIn, routing.Wrap(hs.deleteHomePageCard))
25  }
26

```

图 4-13 卡片管理后端路由注册

4.8.3 设备在线状态检查

设备在线状态检查用于支撑业务门户中的在线设备统计与设置弹窗中的在线/离线提示。如图所示，前端首先从设备列表中提取已登记的 deviceMac 并进行规范化处理，然后构造 InfluxDB 查询语句，查询最近 15 秒内 device_data 测量中是否存在该设备的有效字段记录。只要返回结果中包含对应 device_id，就将该设备加入在线设备列表。设备在线状态检查实现如图 4-14 所示。

```

284  const buildDeviceOnlineCheckQuery = (bucket: string, deviceMacs: string[]): string => {
285      const deviceFilter = buildDeviceFilter(
286          deviceMacs.map((deviceMac) => ({
287              room: '',
288              deviceId: null,
289              deviceMac,
290          })))
291      );
292
293      return `from(bucket: "${bucket}")
294      |> range(start: -15s)
295      |> filter(fn: (r) => r["_measurement"] == "device_data")
296      |> filter(fn: (r) => ${VITALS_FIELD_FILTER_EXPR})
297      |> filter(fn: (r) => ${deviceFilter})`;
298  };
299

```

图 4-14 设备在线状态检查

此外，在线状态判断并非简单读取单次上报结果，而是围绕最近时间窗口内是否仍有有效数据帧进行检测。该设计能够避免瞬时空帧或网络轻微抖动带来的误判，使业务门户中的在线统计更符合设备真实运行状态。

4.9 本章小结

本章分别对“毫米波雷达感知与传输链路”“多组件协同时序数据处理架构”与“基于 Grafana 的业务化适配重构”进行了详细设计与实现。综合来看，本章工作具有以下三方面特征：第一，在设备端层面完成了从原始报文到业务字段的完整解析与可靠上报；第二，在平台层面构建了适配毫米波雷达时序特性的数据接入与存储架构，并通过缓存代理机制平衡了实时性与数据库负载；第三，在业务侧基于 Grafana 完成了业务门户、卡片管理、设备管理、风险弹窗与监护数据服务的定制构建。三条主线相互支撑，形成了从感知到业务的完整工程闭环，为第五章的系统测试提供了实现基础。

第五章 系统测试

本章将结合系统实际运行结果，对前文完成的整体实现进行测试与验证。测试工作的重点，并不只是功能是否“能够运行”，更关注系统在真实监护场景中的持续工作能力与工程可落地性：第一，毫米波雷达感知与传输链路的稳定性与连续性；第二，多组件协同时序数据处理架构在多设备并发下的接入、存储与查询能力；第三，基于 Grafana 的业务化适配重构在多设备管控、风险预警与配置管理上的业务支撑能力。

5.1 测试环境

系统测试环境包括硬件环境和软件环境两部分，如表 5-1 所示。硬件环境方面，系统以 LD6002 系列毫米波雷达与 ESP32-S3 开发板为核心采集单元，通过路由器接入无线网络，并在测试终端上观察业务门户、数据库结果与日志信息。软件环境方面，平台采用 Docker Compose 统一编排 Mosquitto、InfluxDB、Grafana、Redis 以及消息订阅程序，后台逻辑分别由 Go 语言扩展接口和 Python 脚本完成，从而保证测试环境与实际部署形态保持一致。

测试过程中，本文采用“设备日志—MQTT 消息—InfluxDB 字段—门户卡片”四级交叉验证方法。该方法能够从采集、传输、存储和展示四个环节同步观察同一事件，便于快速定位问题来源，也使测试结论更贴近系统在真实连续运行状态下的表现。

表 5-1 系统测试环境

类别	配置
雷达设备	LD6002 系列毫米波雷达
边缘控制器	ESP32-S3
消息代理	Mosquitto
时序数据库	InfluxDB 2.x
可视化平台	Grafana（二次开发版）
数据处理脚本	Python 3
部署方式	Docker Compose +nginx

5.2 功能测试方案

5.2.1 设备端功能测试

设备端测试重点验证以下内容：

- (1) 串口能否正确接收并解析跌倒状态帧；
- (2) 人体存在状态变化能否正确识别；
- (3) WiFi 连接失败时是否能切换到 SoftAP 配网模式；
- (4) MQTT 发布是否按 1 秒周期持续进行；
- (5) 发布 JSON 是否包含必要字段。

5.2.2 云端入库测试

云端测试重点验证：

- (1) MQTT 消息是否能被订阅程序接收；
- (2) JSON 与十六进制报文是否能正确解析；
- (3) InfluxDB 中是否生成 device_data 测量；
- (4) 字段是否能够被 Grafana 正常查询。

5.2.3 业务门户测试

业务门户测试包括：

- (1) 业务门户是否能正确显示各类设备卡片；
- (2) 风险颜色、趋势箭头和陈旧数据灰显是否正常；
- (3) 设备类型筛选与分页是否正常；
- (4) 手动刷新是否能更新数据；
- (5) 点击卡片是否可跳转对应仪表板。

5.2.4 管理模块测试

设备管理与卡片管理模块测试包括：

- (1) 新增设备后是否可在列表中显示；
- (2) 修改设备类型后卡片是否能同步更新；

- (3) 删除设备后是否能同步解除卡片关联；
- (4) 新增卡片后业务门户是否立即反映新配置；
- (5) 是否正确阻止一台设备绑定多张卡片。

5.2.5 监护数据服务测试

监护数据服务测试重点验证：

- (1) 监护页面请求是否能够正确返回指定时间段的心率、呼吸率和跌倒状态数据；
- (2) 风险提示逻辑是否能根据跌倒持续时间和人体在位状态触发相应标记；
- (3) 设备在线检测结果是否能与最新上报时间保持一致；
- (4) 固定时段回放与实时数据合并后，页面展示是否仍保持连续和稳定；
- (5) 查询接口在不同时间表达和设备映射条件下是否能够返回正确结果。

5.3 关键测试用例

5.3.1 跌倒状态上报测试

本用例用于验证跌倒事件从边缘感知端生成、经 MQTT 传输、写入时序库并在业务门户告警卡片中呈现的全链路一致性。测试过程中，通过触发雷达跌倒状态帧与恢复状态帧，分别观察串口解析日志、主题消息载荷、InfluxDB 对应字段取值以及门户颜色与持续计时变化，检查事件标识在多层模块间是否保持同步。

测试结果表明，跌倒状态出现后，设备端能够及时完成状态解析并上报，云端订阅程序可在同一采样周期内完成入库处理，门户卡片随之切换为风险提示状态；恢复正常后，页面展示与数据库记录同步回退，表明该链路具备较好的实时性与一致性。

5.3.2 设备离线检测测试

本用例用于验证系统对设备在线状态的识别能力。测试方法为在设备正常上报一段时间后，主动暂停 MQTT 发布或断开网络连接，并持续观察最近时间窗

口内的最新数据时间、在线设备数量统计以及业务门户设备状态标识，检查离线判定逻辑是否能够按照设定阈值生效。

测试结果表明，当设备在设定时段内未产生新数据时，系统能够将对应设备标记为离线，并同步更新业务门户统计信息。恢复网络连接后，设备状态可随新数据到达而重新转为在线，说明该机制能够满足连续监护场景下的设备可达性检测需求。

5.3.3 设备配置一致性测试

本用例用于验证关系型配置数据与监测数据之间的映射关系是否稳定。测试过程中，对设备名称、设备类型、MAC 地址及所属房间等配置项进行新增与修改，随后检查设备管理列表、卡片绑定下拉项以及门户卡片标识是否同步更新，并核对历史监测数据是否仍能够正确关联到目标设备。

测试结果表明，设备配置变更后，系统能够保持展示层、配置层与数据层的一致性。其中，MAC 地址规范化处理能够减少人工录入差异带来的匹配错误，设备类型调整后业务门户也能按对应模板完成渲染，说明配置管理模块具备较好的可维护性。

5.3.4 卡片配置联动测试

本用例用于验证门户卡片配置对监护门户展示效果的联动能力。测试时分别修改房间名称、绑定设备、显示标题和仪表板 UID 等关键字段，保存后刷新业务门户，并进一步检查卡片内容、跳转链接及页面布局是否按最新配置立即生效。

测试结果表明，卡片配置保存后即可驱动业务门户采用新参数重新渲染，卡片标题、监测对象与仪表板跳转目标均与后台配置保持一致。该结果说明系统已将监护业务门户由静态页面扩展为可配置门户，便于后续按照养老房间与设备分布进行灵活调整。

表 5-2 关键测试用例

编号	测试内容	输入/操作	预期结果
T1	设备端跌倒状态上报	模拟跌倒触发	MQT 收到 fall=1 , fall_count 开始累加
T2	人体存在状态识别	进入/离开雷达区域	云端接收到 human=1/0 变化
T3	数据入库	设备连续上报	InfluxDB 中出现对应 字段记录
T4	业务门户加载	打开业务门户	卡片列表正常显示, 支 持筛选分页
T5	设备管理	新增一个设备	设备列表增加记录, 可 供卡片绑定
T6	卡片管理	新增卡片并绑定仪 表板	业务门户增加新卡片 并可点击跳转
T7	在线检测	设备断网	在线统计数下降, 设备 显示离线
T8	跌倒告警	跌倒持续超过阈值	业务门户弹出风险窗 口

5.4 测试结果分析

5.4.1 功能闭环分析

综合上述测试结果可见, 本文构建的系统已形成较完整的功能闭环: 设备端能够稳定输出跌倒状态与人体在位状态; 云端能够完成多设备并发数据接入与时间序化管理; 业务门户能够展示多类设备卡片并支持配置管理; 监护数据服务能够围绕核心监测指标提供统一查询与状态组织能力。具体而言, 设备管理与卡片管理模块提升了系统的可维护性, Grafana 业务化适配重构增强了监测信息展示的完整性, 监护数据服务的引入则使系统由单一可视化平台进一步发展为具备管理与联动能力的综合监测平台。

5.4.2 链路稳定性分析

从链路完整性看，设备端状态生成、消息队列传输、时序数据落库、门户页面展示与监护数据服务调用之间已形成较稳定的闭环关系。各层职责划分较为清晰，当单一环节发生短时波动时，其余模块仍可通过缓存、回放或“最后一次有效值”等机制维持页面可用性。这种容错特性对养老监护场景下的连续观察尤为重要。

5.4.3 系统局限性与改进方向

尽管系统满足课题设定的核心功能要求，但仍存在若干进一步优化空间：

- （1）多设备环境下的自动设备发现机制仍相对基础，目前依赖人工添加设备 MAC 信息；
- （2）风险判断规则以阈值与统计指标为主，尚未引入更完善的长期趋势建模方法；
- （3）当前监护数据服务主要围绕少数核心指标展开，后续仍可扩展更多状态分析与业务联动能力；
- （4）跨房间多设备场景下的批量配置仍较为繁琐，可进一步引入模板化与导入/导出机制。

5. 本章小结

本章从设备端、云端、业务门户、配置管理与监护数据服务五个层面对系统进行了系统化测试。综上，系统在功能闭环、链路稳定性与业务适配能力三方面均达到了课题设定的目标，具备较好的工程完整性与应用可行性。

第六章 总结与展望

6.1 论文总结

本文围绕智慧养老健康监测场景下的连续监护需求,针对当前面向养老监护的毫米波雷达系统在“端到端一体化系统实现”“适配毫米波雷达时序特性的数据接入与存储方案”以及“适配业务需求的可视化展示与管理载体”等方面的不足,完成了一套面向养老监护的毫米波雷达人体运动理解系统的设计与实现。论文首先分析了智慧养老连续监测的现实需求,指出非接触式监测与平台化监护的重要性;随后围绕系统需求、总体架构、模块设计与详细实现进行了系统阐述;最后通过功能测试与关键测试用例对系统进行了验证。整体而言,本文的研究工作可归纳为以下三方面成果:

(1) 毫米波雷达感知与传输链路的构建

传统接触式监测方案依从性不足、连续性不强的问题,本文构建了基于 LD6002 毫米波雷达与 ESP32-S3 边缘控制器的感知与传输链路。该链路实现了雷达串口报文解析、跌倒状态识别、有人/无人状态提取、WiFi 联网、SoftAP 配网与 MQTT 数据上报,完成了监护数据的边缘感知与可靠传输,从根本上解决了传统方案在非接触、连续性部署方面的不足。

(2) 面向监护场景的多组件协同时序数据处理架构

针对现有养老监护系统缺乏适配毫米波雷达时序特性的数据接入与存储方案的问题,本文提出了一种面向监护场景的多组件协同时序数据处理架构。该架构基于 Mosquitto 构建多设备并发数据接入层,针对居家环境下雷达数据高频、易丢包的特点优化 MQTT 消息分发与缓存策略;基于 InfluxDB 设计适配生命体征与状态数据的时序存储模型,针对心率、呼吸、跌倒状态等监护数据的特性定制存储字段与索引结构;并通过缓存代理与历史回放机制兼顾实时性与稳定性。该架构为后续监护业务功能开发提供了高可靠的数据基础。

(3) 基于 Grafana 的养老监护业务化适配重构

针对现有养老监护系统缺乏适配业务需求的可视化展示与管理载体的问题,本文在 Grafana 框架基础上开展了面向养老监护场景的业务化适配重构。重构工

作定制构建了业务门户、多类型卡片差异化渲染、设备管理、卡片管理、风险弹窗告警、设备在线检测与分页筛选等场景专属管控业务模块，并通过统一的监护数据服务接口将底层时序数据组织为前端可直接消费的结构化结果。真实居家场景的测试结果表明，该系统能够完成跌倒状态与人体在位状态的稳定上报，验证了毫米波雷达、物联网时序数据平台与养老监护业务系统结合的可行性，为智慧养老健康监护系统的工程落地提供了参考。

总体而言，本文所构建的系统已由单纯的“数据展示原型”发展为具备门户展示、配置管理与监护数据服务能力的综合监测系统，较好地兼顾了工程实现深度与综合集成广度。

6.2 未来展望

在上述研究与实现基础上，后续研究可从以下若干方向继续完善与拓展：

（1）在人体运动理解层面，可在现有跌倒检测与生命体征识别能力的基础上，进一步结合更丰富的时序特征与轻量化序列建模方法，提升复杂动作分类、长时行为模式识别与早期风险预警能力；

（2）在监护数据服务层面，可扩展更多健康指标（如睡眠分期、长期心率变异性、呼吸节律变化）与复杂场景下的状态判定规则，并引入基于历史数据的个性化基线模型以提高风险判断的特异性；

（3）在系统架构层面，可在保留现有 Mosquitto + InfluxDB + Grafana 技术栈的前提下，探索接入更多类型传感器（如毫米波多目标识别、低分辨率红外阵列、压电式床垫等）并通过同一监护数据服务对外暴露，进一步增强系统的可扩展性；

（4）在门户展示层面，可在卡片化业务门户之外补充面向护理人员与家属的多端入口（如移动端 H5、小程序），并增强数据解释深度与场景化表达，提高监护结果的可读性；

（5）在系统治理层面，可继续完善角色权限、操作审计、配置版本管理与租户级隔离机制，并引入面向养老机构合规需求的数据脱敏与导出策略；

(6) 在评测体系层面,可进一步在更多真实居家场景下开展长周期对照测试,并与穿戴式设备的同步采集结果进行横向比较,以更系统地评估毫米波雷达方案在养老监护场景下的工程优势与边界条件。

综合而言,本文所构建的系统为面向养老监护的毫米波雷达人体运动理解系统提供了一个可工程落地的参考实现,并为后续在智慧养老健康监测领域的深入研究保留了较为开放的扩展空间。

参考文献

- [1] MA R, ZHANG X Y. Indoor Human Activity and Vital Sign Monitoring Based on Millimeter Wave Radar[C]//2022 Cross Strait Radio Science & Wireless Technology Conference (CSRSWTC). Haidian, China, 2022: 1-1.
- [2] JANG M H, KANG S W, LEE S. Monitoring Person on Bed Using Millimeter-Wave Radar Sensor[C]//2022 IEEE Radar Conference (RadarConf22). New York City, NY, USA, 2022: 1-4.
- [3] ZHENG Z, HUO J, WANG J. Millimeter-Wave Radar-Based Vital Sign Signal Extraction Using Multiple Distance Bin[C]//2024 International Applied Computational Electromagnetics Society Symposium (ACES-China). Xi'an, China, 2024: 1-3.
- [4] LIU K, et al. UniSleepPos: Sleep Posture Identification System Utilizing Millimeter-wave Radar[C]//2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Lisbon, Portugal, 2024: 2232-2237.
- [5] HUANG T, LIU G, LI S, et al. 基于毫米波雷达的人体活动识别系统: RPCRS[C]// 2022 年 IEEE 第 28 届国际并行与分布式系统会议(ICPADS)论文集. 南京: IEEE, 2022: 122 - 129.
- [6] TANG H, et al. GF-DecNet: Geometry-Aware Feature Decoupling Network for Millimeter-Wave Radar Pose Estimation[J]. IEEE Internet of Things Journal, 2025, 12(21): 45596-45609.
- [7] BUCCAFURRI F, DE ANGELIS V, LAZZARO S. MQTT-I: Achieving End-to-End Data Flow Integrity in MQTT[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(05): 4717-4734.
- [8] DOSHI R, INAMDAE S, KARMARKAR T, et al. DistributedMQTTBroker: A Load-Balanced Redis-Based Architecture[C]//2024 International Conference on Emerging Smart Computing and Informatics (ESCI). Pune, India, 2024: 1-6.
- [9] BOUBLENZA R N, BENADDA B, BORSALI A R. Hardware Design of a Low-cost IOT Current Monitoring with MosquittoMQTTBroker[C]//2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC). Setif, Algeria,

2024: 1-3.

[10] AGGOUNE A, BENRATEM Z. ECG Data Visualization: Combining the power of Grafana and InfluxDB[C]//2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAEECS). BLIDA, Algeria, 2023: 1-6.

[11] ZHU X, NIE X, LIU J. Time Series Database Optimization Based on InfluxDB[C]//2023 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEEC). Athens, Greece, 2023: 879-885.

[12] HINDI I, ALYAMAN M, ABOZENAH A, et al. Smart Alarm IoT System: Monitoring Elevator Traffic and Meteorological Data on Job Sites UsingMQTTand InfluxDB integrated with Grafana[C]//2024 15th International Conference on Information and Communication Systems (ICICS). Irbid, 2024.

[13] RAHMAN M A, LI Y, NABEED T, et al. Remote monitoring of heart rate and ECG signal using ESP32[C]//2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE). Changsha, China, 2021: 604-610.

[14] SEREPAS F, PAPIAS I, CHRISTAKIS K, et al. Lightweight Embedded IoT Gateway for Smart Homes Based on an ESP32 Microcontroller[J]. Computers, 2025, 14: 391.

[15] 杨光, METCALFE B W, WATSON R J, 等. 基于毫米波雷达相位连续性与信号质量指数的驾驶员非接触式心肺监测系统[J]. IEEE 物联网期刊, 2025, 12(20): 43480-43494.

[16] TAKIMOGLU. 30 Internet of Things – IoT stats from reputable sources in 2021[EB/OL]. [2022-01-27].

[17] YANG G, METCALFE B W, WATSON R J, et al. A noncontact cardiorespiratory monitoring system for drivers using millimeter wave radar with phase continuity tracking and signal quality index[J]. IEEE Internet of Things Journal, 2025, 12(20): 43480–43494.

-
- [18] KANG W, KE F, CHANG Y C. Non-contact realtime vital signs monitoring system based on millimeter wave FMCW radar[C]// 2023 9th International Conference on Computer and Communications (ICCC). Chengdu, China: IEEE, 2023: 1594–1599.
- [19] CUI H, DAHNOUN N. High precision human detection and tracking using millimeter-wave radars[J]. IEEE Aerospace and Electronic Systems Magazine, 2021, 36(1): 22–32.
- [20] QIAO X, SU Y, LI X, et al. Millimeter-wave radar vital signs measurement with random body movement using missing data model[J]. IEEE Transactions on Instrumentation and Measurement, 2025, 74: 1–14.
- [21] MASHHADIGHOLAMALI M, et al. FMCW radar-based human activity recognition: a machine learning approach for elderly care[C]// 2025 IEEE Wireless Communications and Networking Conference (WCNC). Milan, Italy: IEEE, 2025: 1–6.
- [22] ISLAM S M M, LUBECKE V M. Sleep posture recognition with a dual-frequency microwave Doppler radar and machine learning classifiers[J]. IEEE Sensors Letters, 2022, 6(3): 1–4.
- [23] NAIK N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP[C]// 2017 IEEE International Systems Engineering Symposium (ISSE). Vienna, Austria: IEEE, 2017: 1–7.
- [24] SHEN Z, NUNEZ-YANEZ J, DAHNOUN N. Multiple human tracking and fall detection real-time system using millimeter-wave radar and data fusion[C]// 2023 12th Mediterranean Conference on Embedded Computing (MECO). Budva, Montenegro: IEEE, 2023: 1–6.
- [25] LIN F, et al. SleepSense: a noncontact and cost-effective sleep monitoring system[J]. IEEE Transactions on Biomedical Circuits and Systems, 2017, 11(1): 189–202.

- [26] ZHOU T, XIA Z, WANG X, et al. Human sleep posture recognition based on millimeter-wave radar[C]// 2021 Signal Processing Symposium (SPSymposium). Lodz, Poland: IEEE, 2021: 316–321.
- [27] KITTIYANPUNYA C, CHOMDEE P, BOONPOONGA A, et al. Millimeter-wave radar-based elderly fall detection fed by one-dimensional point cloud and Doppler[J]. IEEE Access, 2023, 11: 76269–76283.
- [28] WU Y, et al. mmHPE: robust multiscale 3-D human pose estimation using a single mmWave radar[J]. IEEE Internet of Things Journal, 2025, 12(1): 1032–1046.

致谢

本论文自选题、调研、设计、实现、测试到论文撰写的整个过程，倾注了诸多师长、同学与家人的关心和帮助，在此谨向他们致以最诚挚的谢意。

首先，衷心感谢我的指导教师 in 课题选题、技术路线确定、方案细节讨论与论文修改各阶段所给予的悉心指导与耐心包容。导师严谨求实的治学态度、对工程实现细节的认真把握以及对智慧养老应用场景的深刻理解，使我在面对复杂系统设计 with 多组件协同问题时能够保持清晰的思考路径，对我此后的学习与工作产生了深远影响。

感谢实验室同学在系统联调、设备部署与测试过程中给予的无私协助，与各位同学的讨论让我在毫米波雷达数据处理、InfluxDB 时序建模、Grafana 二次开发等具体技术问题上少走了许多弯路。

感谢我的家人在学习生涯中给予的长期支持与理解，他们的关爱是我面对科研挑战时最坚实的后盾。

最后，向在百忙之中评阅本论文的各位专家、老师致以诚挚的谢意。文中疏漏在所难免，敬请批评指正。



华侨大学教务处

泉州校区地址：

福建省泉州市丰泽区城华北路269号

邮编：362021

厦门校区地址：

福建省厦门市集美区集美大道668号

邮编：361021

主页：<https://jwc.hqu.edu.cn/>