杨萍, 房可佳. 组态式智能家居人机交互系统的设计与实现[J]. 智能计算机与应用,2025,15(2):153-161. DOI:10.20169/j.issn.2095-2163.250224

组态式智能家居人机交互系统的设计与实现

杨 萍, 房可佳
(陕西科技大学 电子信息与人工智能学院, 西安 710021)

摘 要: 智能家居的人机交互系统影响用户体验,而传统的菜单式的交互页面存在着画面固定、操作不直观等问题,并且系统通用性较差,实时性也不高。针对上述问题,本文开发了一种基于云平台的组态式智能家居人机交互系统,设计了包括页面、数据和交互控制组件在内的组件,以此为基础实现了图形化的界面构建。针对智能家居设备数据多源异构的特点,采用 MongoDB 和 Redis 相结合的存储方案进行分类存储,并以 Redis+WebSocket 的方式实现数据的实时推送。实验表明,系统功能正确,可实现智能场景搭建、设备交互,实时更新设备数据及统计能耗。

关键词: 智能家居; 人机交互系统; 云平台; 组态式; 实时推送
中图分类号: TP302.1 **文献标志码**: A **文章编号**: 2095-2163(2025)02-0153-09

Design and implementation of a configured smart homehuman-computer interaction system

YANG Ping, FANG Kejia
(School of Electronic Information and Artificial Intelligence, Shaanxi University of Science and Technology, Xi′an 710021, China)

Abstract: The human-computer interaction system of the smart home is very important in affecting the user experience, while the traditional menu-style interactive page has the problems of fixed screen and unintuitive operation, and the system has poor versatility and real-time performance. Aiming at the above problems, this paper develops a cloud platform-based configuration smart home human-computer interaction system, designs components including pages, data and interactive control components, and realizes the graphical interface construction based on this . In view of the multi-source and heterogeneous characteristics of smart home device data, a storage solution combining MongoDB and Redis is used for classified storage, and real-time data push is realized in the form of Redis+WebSocket. Experiments show that the system functions correctly, and can realize intelligent scene construction, device interaction, and real-time update of device data and statistical energy consumption.

Key words: smart home; human-computer interaction system; cloud platform; configuration; real-time push

0 引 言

随着科技的快速发展,智能家居已经成为人们日常生活中不可或缺的一部分[1]。作为智能家居的重要组成部分,人机交互系统与用户的使用体验密切相关[2]。

目前的智能家居控制多以 APP、小程序提供层层筛选的菜单式控制页面,操作繁琐、不直观,且往往需要用户下载多个应用才能实现多个互联网公司的设备控制。为满足用户多样化、定制化的需求,本文借鉴组态的思想,采用图形可视化方式,将系统以组件形式进行配置和组织,从而实现灵活、可扩展的开发。另一方面,传统的组态化平台大多采用客户机/服务器(C/S)模式进行开发,适用面较窄,数据实时性不佳,维护成本高,拓展性也较低[3-4]。相比之下,浏览器/服务器(B/S)模式能够提高数据的集成化程度,实现运行和控制数据的整合,进一步实现对智能家居人机交互系统功能的集成与整合[5-6]。

本文采用 B/S 架构,开发一个智能家居通用的、交互式的云组态平台,通过简易拖拉拽可视化的

基金项目: 陕西省重点研发计划项目(2023-YBGY-208)。
作者简介: 杨 萍(1979—),女,硕士,副教授,主要研究方向:计算机应用技术,边缘智能计算,人工智能等。
通信作者: 房可佳(1998—),女,硕士研究生,主要研究方向:计算机应用技术。Email:211612110@sust.edu.cn。
收稿日期: 2023-08-17

智能家居设备模型控件,可快速构建智能家居场景,以直观、动态的形式展示现实世界家居设备的空间分布、连接关系、运行状况和统计数据。

1 智能家居系统架构

1.1 系统架构

系统采用如图 1 所示的“云-边-端”架构为基础架构。云侧是用户对设备进行远程管控的入口;边侧采用智能网关设备统一接入,管理家居设备;端侧由各类家居设备构成。系统综合应用物联网、云组态等多项技术与手段,对智能家居设备的状态和基本参数等数据进行全面整合。

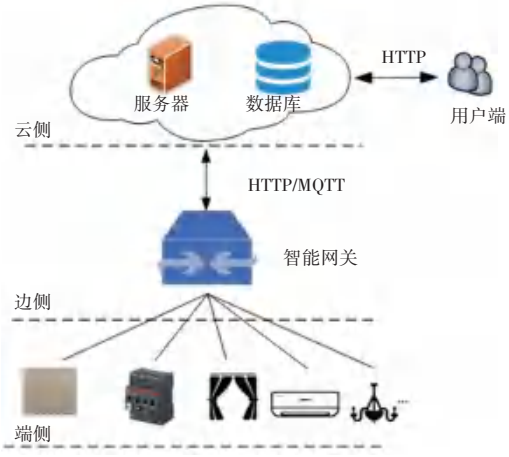

图 1 系统架构

Fig. 1 System structure

(1)“云”侧:是整个系统的核心,主要实现对设备的远程管理和控制,对外提供 RESTful (Representational State Transfer)风格的接口,用户可通过这些接口访问服务器,实现对设备的交互控制;

(2)“边”侧:由于家居设备的种类繁多,接口也不相同,因此采用智能网关设备来实现对家居设备的统一接入,该网关向下提供对家居设备的管控,向上提供联网、与云服务器交互;

(3)“端”侧:集成多种终端子设备,如控制器、照明灯具、视频监控等,可接收来自边侧智能网关设备的控制命令,修改自身属性,实现智能家居系统控制的最后一环。

1.2 功能需求

针对智能家居系统目前存在的问题,结合对云组态技术和系统架构的分析,将云平台的功能分为系统管理、模型交互、设备配置和数据展示 4 个模块,各模块内部业务划分及具体功能如图 2 所示。

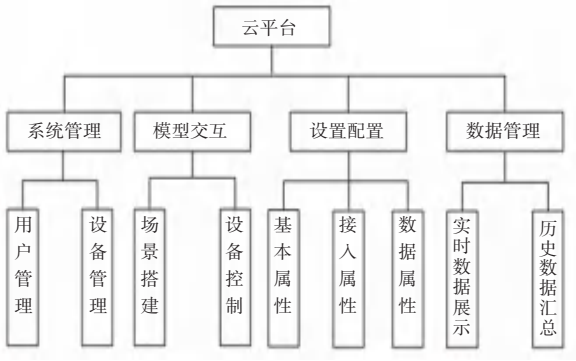

图 2 系统功能结构图

Fig. 2 System function structure diagram

(1)系统管理:分为用户管理和设备管理。包括用户登录、注册、修改密码;设备的接入、增加、修改、删除、绑定等操作。

(2)模型交互:用户可以通过在云组态界面简易的拖拉拽模型控件搭建起属于自己的家庭场景,并实现对场景中可控设备的操作。

(3)设备配置:配置设备的相关信息,包含设备基本属性、接入属性和数据属性。

(4)数据展示:接收边缘网关上报的设备实时数据并主动推送至页面,汇总设备的历史数据,生成相关的统计结果。

1.3 组态平台方案设计

智能家居组态平台架构如图 3 所示,由组态编辑系统、组态运行系统、组态描述文件、数据库服务器和 MQTT(Message Queuing Telemetry Transport)服务器 5 部分组成。组态编辑系统由各类组件构成,完成对智能家居场景的搭建,包括场景布局构建和设备配置,形成描述组态页面和模型配置信息的组态描述文件,通过组态描述文件连接组态编辑系统和组态运行系统。组态运行系统通过对组态描述文件的解析,加载相应的组件重建智能家居场景。

针对智能家居系统中设备种类众多,数据类型多样的特点,为提高数据操作的安全性,减少并发操作时出现的数据冲突问题,将系统中的数据分类存储,并且为不同类型的数据开放不同的接口。使用扩展性好的文档型数据库 MongoDB 来存储系统基本信息和设备历史数据,其数据结构较为松散,适合系统中多源异构数据的存储。为提高数据实时性,利用 Redis 内存数据库读取快的优点,存储设备实时数据[7-8]。采用 EMQ(Erlang/Enterprise/Elastic MQTT Broker)作为 MQTT 服务器,实现与边缘网关的通信,进而实现与系统中子设备的交互控制。

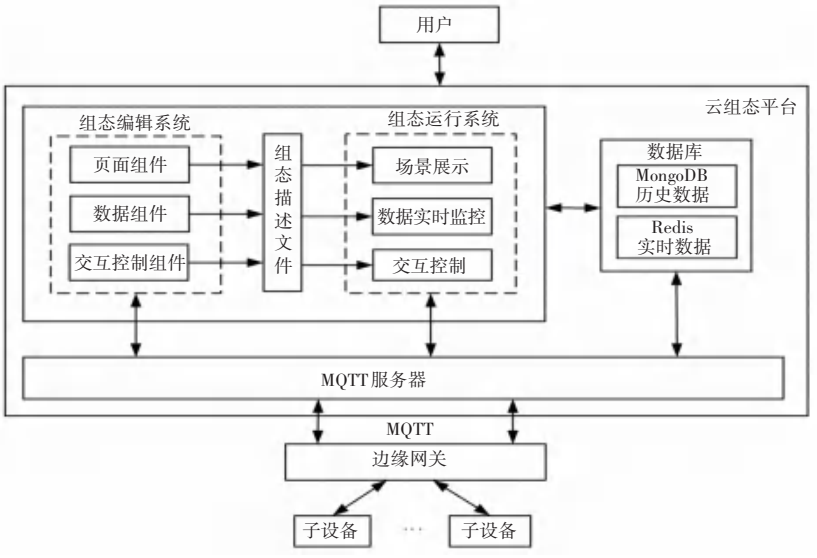

图 3　智能家居组态平台架构

Fig. 3　Smart home configuration platform architecture

2　智能家居组态平台设计与实现

2.1　组件模型设计

组件是一种抽象模型，具有明确定义的接口，封装了实现特定功能所需的信息[9-10]。组件模型如图 4 所示，每个组件都包含两个基本元素：组件接口(I)，组件属性(A)。组件接口 I 包含用于对外提供组件功能的 I_F 接口和表示组件之间关联关系的 I_C 接口。属性描述组件的状态或配置信息，这些属性可以影响组件的行为和功能，通过组件接口可以访问或修改这些属性。

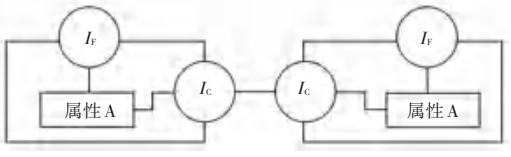

图 4　组件模型

Fig. 4　Component model

对于智能家居系统，主要的组成元素有设备控件、房间结构控件、图表控件、数据库和设备交互控制命令等。通过对这些组成元素的分析归纳，将构成智能家居组态平台的元素按照功能划分为 3 类：页面组件、数据组件和交互控制组件。

2.1.1　页面组件

页面组件是人机交互系统在屏幕上的表现形式，以图形化方式来构建智能家居场景是组态式人机交互系统与传统编码式构建人机交互系统的最大区别[11-12]。对于智能家居系统来说页面组件可分为模型组件和交互控制组件两种，如图 5 所示。模型组件包括房间结构组件和设备组件两种，用于搭建智能家居场景。根据智能家居系统中设备控制方式的不同，可将家居设备分为开关类、可调控类、数值类和菜单类，并为不同种类的设备提供不同的交互组件。

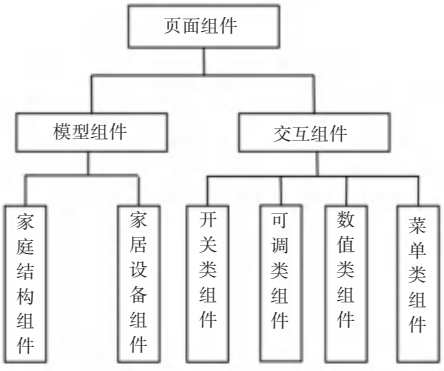

图 5　页面组件

Fig. 5　Page components

2.1.2　数据组件

设备数据是支撑智能家居系统运行的关键组成部分，由于设备种类繁多且通信协议接口各异，因此云组态平台的首要考虑是构建一个灵活的设备数据组件[13-14]。

属性作为设备数据的基本单位，以键-值对的形式表示设备特征。基于属性的方法能够将设备看作属性列表，通过更改属性值来实现设备状态控制[15]。对于智能家居设备，可将属性分为基本属性、接入属性和数据属性 3 类，如图 6 所示。

(1)基本属性：设备 ID、设备名称、生产厂家和设备在线状态等描述设备基本信息的属性；

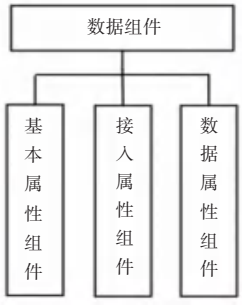

图 6 数据组件

Fig. 6 Data components

(2)接入属性:接入的边缘网关、接入协议、协议内容等用于发现子设备的信息;

(3)数据属性:分为静态配置数据和动态状态数据,静态配置数据在设备运行期间很少更改,而动态状态数据经常更改并表示设备的当前状态。

为了方便设备属性的扩展,为各类设备属性提供统一的属性描述结构,根据表 1 所示,主要包括数据的各类控制属性,以及数据存储与报警的相关参数。

表 1 属性描述结构

Table 1 Attribute description structure

参数描述	数据类型
数据属性标识	String
数据种类,基本/接入/动态/静态数据	String
数值类型,如 String/Double/int/bool/enum 等	String
控制类型,分为只读(r)/只写(w)/可读可写(rw)	String
数据属性最大值	String
数据属性最小值	String
数据属性报警值,当数据属性值到达此值时,触发报警并向用户推送相关消息。	String
是否保存历史	Bool
数据采集周期	Int

2.1.3 交互控制组件

交互控制组件如图 7 所示,是系统运行与交互的重要支撑。交互控制组件主要包含两部分功能,一是作为页面组件和数据组件之间的沟通桥梁,实现组态页面上设备数据的更新,主要包含两种实现方式:数据主动推送和定时轮询;二是实现系统和用户之间的逻辑控制,如用户通过页面交互组件对设备下发控制命令,改变设备属性,或是对报警事件等系统消息的响应和控制。

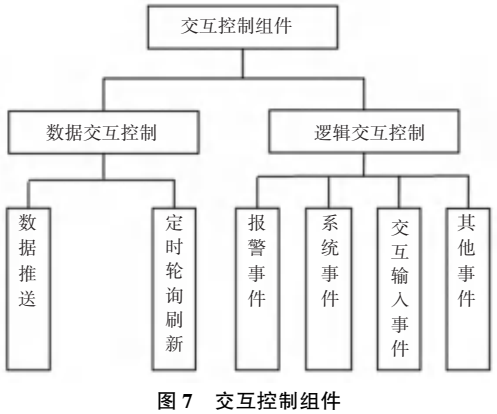

图 7 交互控制组件

Fig. 7 Interactive control components

2.2 组态编辑文件的格式

组态编辑文件是一种为系统运行提供基础描述信息的配置文件,为组态运行系统的正确运行提供基础信息。组态描述文件结构如图 8 所示,采用 JSON 结构,主要包含系统的项目信息和模型信息。项目信息描述用户创建的当前项目的 ID 和名称;模型信息分为页面信息和模型绑定信息两部分,页面信息来自于页面组件,包括构成场景的模型组件和模型组件在场景中所处的位置。模型绑定信息由模型组件所绑定的实际设备的设备 ID 和设备所具有的数据属性构成,用于实现组态场景与实际场景的数据交互。

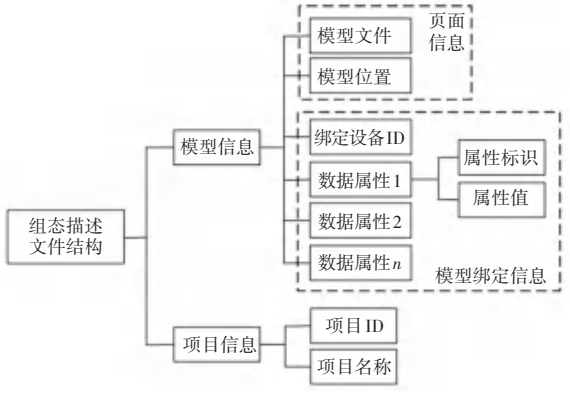

图 8 组态描述文件的结构

Fig. 8 Structure of configuration description file

组态描述文件需要在用户完成在组态编辑系统中的场景搭建后,保存并上传至服务器,由服务器处

理并保存至数据库项目集合下。组态描述文件的生成及保存过程如图 9 所示。

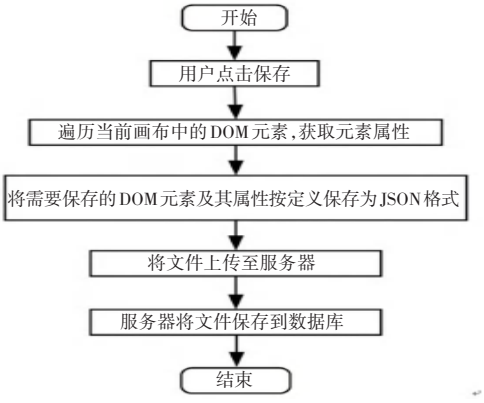

图 9　组态描述文件保存流程图

Fig. 9　Flowchart of saving the configuration description file

通过遍历当前画布上的 DOM(Document Object Model)元素，依次获取每一个元素的属性，将其中需要保存的信息按照定义的文件格式组织并上传至服务器，由服务器处理并保存在数据库中。

2.3　数据库逻辑结构设计

2.3.1　系统数据流

智能家居系统的数据流图如图 10 所示。用户填写注册信息登录进平台，创建项目并配置设备属性，形成设备、人员和项目数据文档集合，并对外提供查询接口。边缘网关根据用户配置的设备属性采集设备信息，将数据打包上传至平台，由平台对数据分类处理并转存至实时数据库和历史数据库中，实时数据由服务器主动推送至组态页面，历史数据经数据统计处理后，对外提供历史数据统计分析接口。

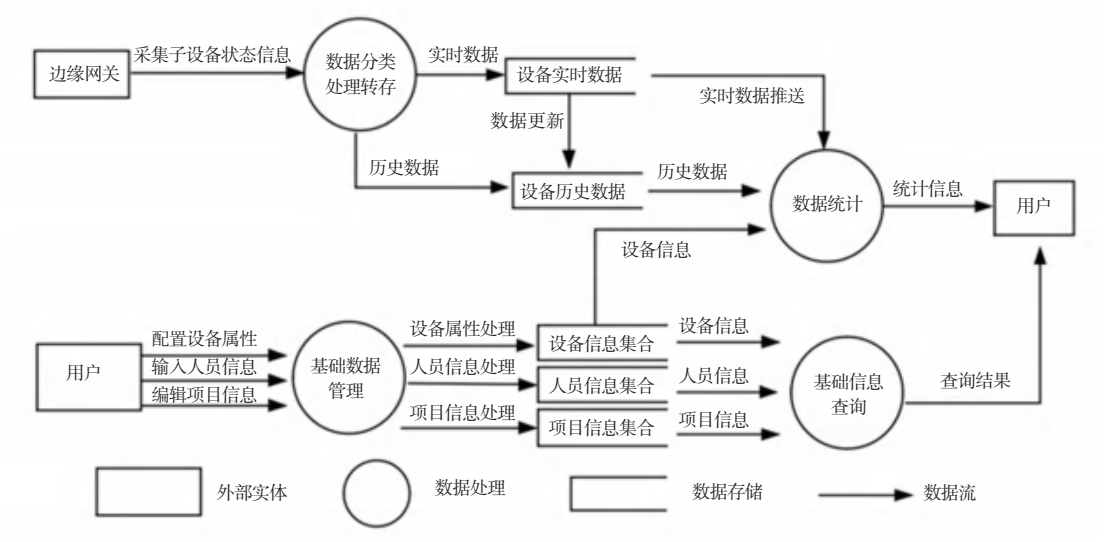

图 10　智能家居系统数据流图

Fig. 10　Smart home system data flow diagram

2.3.2　数据存储模型

智能家居系统中的实时数据以 Key-Value 键值对的形式存储在实时数据库 Redis 中。与传统的以数据点为基本单位进行存储不同的是本系统以设备为基本单位存储，系统中的每一个子设备都有唯一的 ID 值，以“Sub:”字符串拼接子设备 ID 作为 Key 值，Value 值采用 hash 结构，以数据属性标识为 key，值为 value，均为 String 类型。

MongoDB 数据库以文档的形式存储数据，多个文档构成集合。根据对系统数据的分析，历史数据库主要包括的集合有用户、边缘网关、项目和子设备，其中用户集合与项目集合和边缘网关集合为一对多的关系，一个用户可以管理多个项目和边缘网关；子设备集合与边缘网关集合为多对一的关系，一个边缘网关下可接入多个子设备；项目集合与边缘网关集合为一对多的关系，即一个项目可以绑定多个边缘网关。历史数据库存储模型如图 11 所示。

2.4　数据实时更新

传统的数据更新多采用定时轮询的方式，系统定时查询数据库中的设备数据从而将数据更新至页面，这种更新方式对设备数据更新不及时，系统运行效率也不高[16]。

为解决以上问题，本文利用 Redis 的键空间通知功能，实时监听 Redis 中的数据变化。同时为实现数据的主动推送，采用可以向客户端主动推送信息的 WebSocket 协议，数据更新的过程如图 12 所示。MQTT 客户端将设备数据更新到 Redis 服务器后，客户端监听到数据库中键的更新，发起 WebSocket 会话，将数据推送至组态页面。

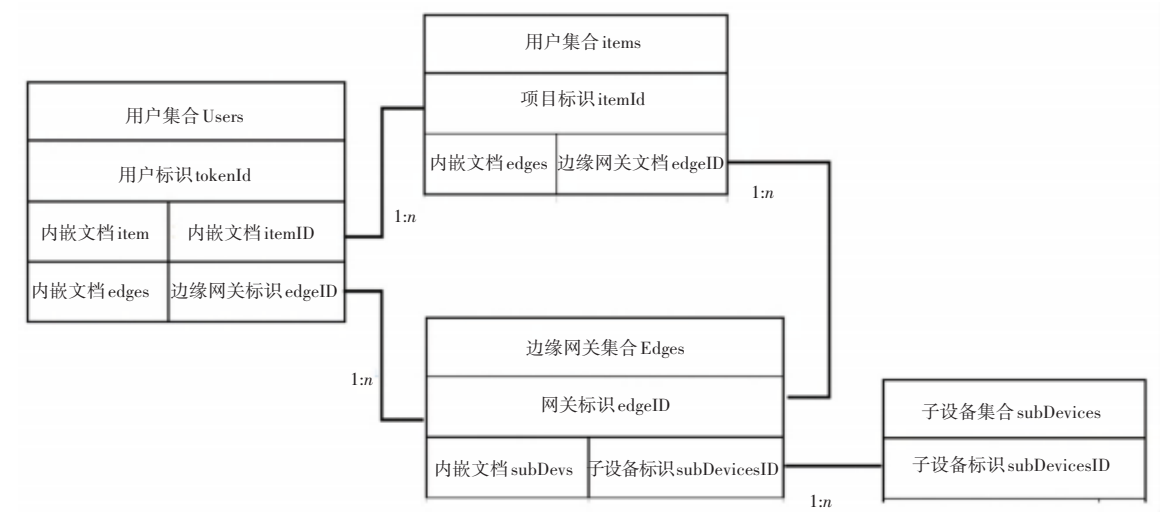

图 11 历史数据库存储模型图

Fig. 11 Historical database storage model diagram

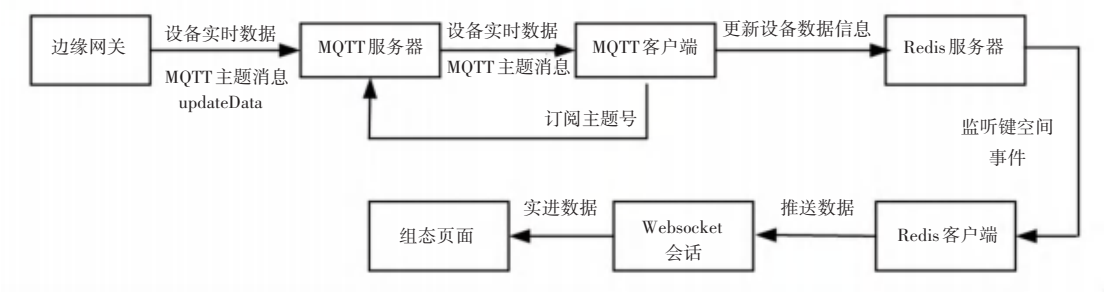

图 12 数据更新过程示意图

Fig. 12 Schematic diagram of the data update process

2.5 历史数据统计分析

利用 Echarts 图表对设备的历史数据进行统计分析的流程图如图 13 所示。通过引用 Echarts 的配置文件并配置 Echarts 图表,从历史数据库读取设备的历史数据,遍历历史数据将数据注入到图表中,形成统计分析结果。

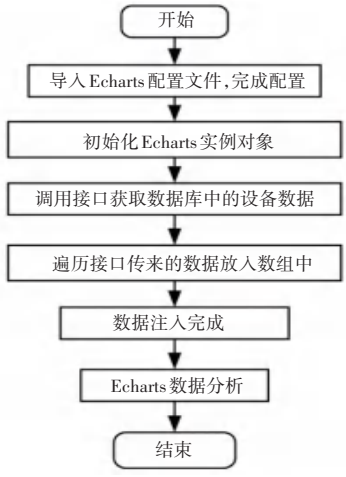

图 13 历史数据统计分析流程图

Fig. 13 Historical data statistical analysis flow chart

3 系统功能验证

3.1 实验平台

为验证系统功能,本文搭建实验平台如图 14 所示。使用 RK3399 控制器作为边缘网关接入组态平台,端侧包含一个由 DALI 网关和 12 个灯具组成的 DALI 子系统以及一个由电能检测设备和 V3s 网关组成的电能检测模块,边缘网关使用 RJ45 接口以 Modbus-Tcp 协议实现与 DALI 网关和 V3s 网关的数据通信。

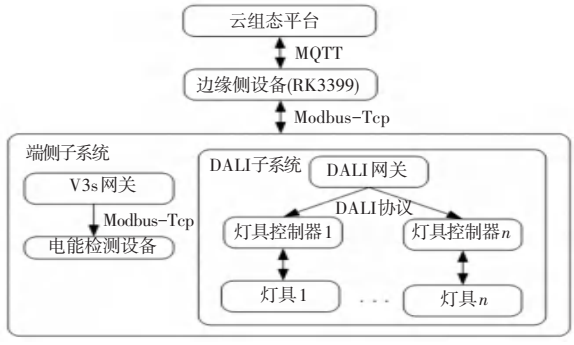

图 14 实验平台示意图

Fig. 14 Schematic diagram of the experimental platform

用户可通过配置接入边缘网关的 DALI 网关和 V3s 网关的 IP 和端口号实现 DALI 网关和电能检测模块的接入；DALI 网关接入后，为 DALI 总线上的灯具分配地址并将子系统中的灯具信息通过边缘网关上报。

3.2　场景搭建与设备交互

最终搭建的组态场景如图 15 所示。通过对 DALI 子系统中灯具的分析，确定灯具所具备的属性见表 2。根据灯具类型的不同，提供不同的交互控制组件，调光调色灯、调光灯和开关灯的交互控制面板如图 16 所示。

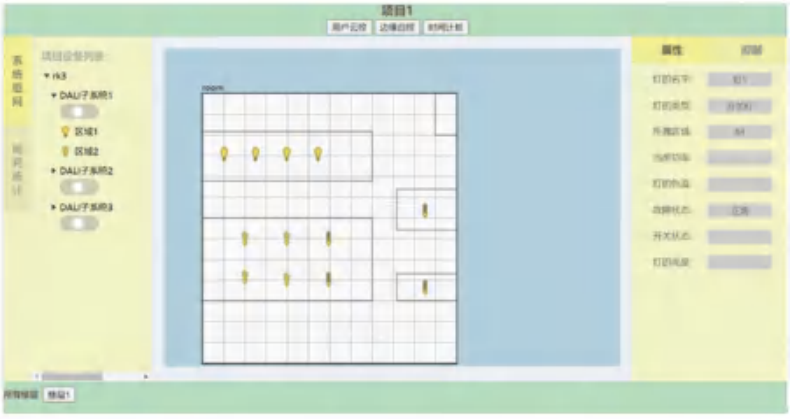

图 15　组态场景搭建结果

Fig. 15　Configuration scene construction result

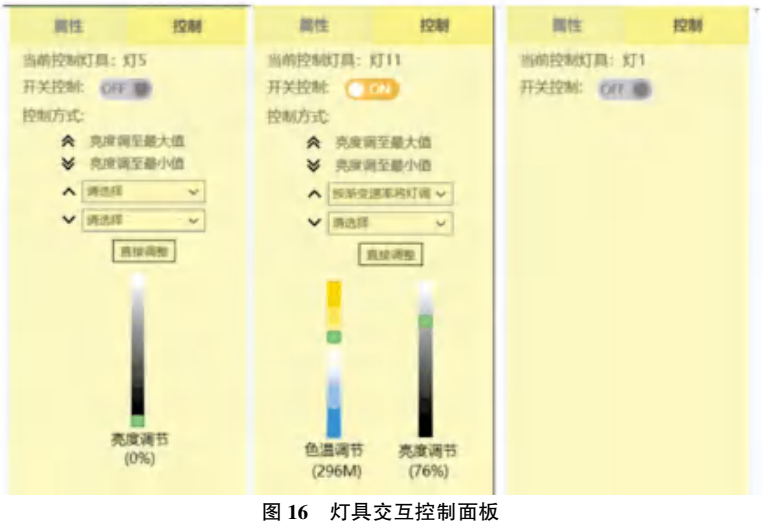

图 16　灯具交互控制面板

Fig. 16　Luminaire interactive control panel

表 2　灯属性列表

Table 2　Lamp property list

属性	名称	数据种类	数值类型	控制类型
ID	短地址	接入属性	int	R
lampType	灯的类型	基本属性	String	R
dalisysAreaSn	区属性(组)	基本属性	int	W/R
controlType	控制方式	基本属性	String	W/R
colourTemp	色温值	动态属性	int	W/R
nowPower	当前功率等级	动态属性	int	W/R
minPower	最小功率等级	静态属性	int	W/R
maxPower	最大功率等级	静态属性	int	W/R
fadeRate	渐变速率	静态属性	int	W/R
fadeTime	渐变时间	静态属性	int	W/R
state	状态属性	动态属性	String	R
LUXpercent	灯的亮度百分比	动态属性	int	W/R
openOff	开关属性	动态属性	Bool	W/R

3.3 通信与数据实时更新功能

对于灯具属性中如亮度百分比、当前功率和色温等动态状态数据,由边缘网关采集这些实时数据并打包上报至平台,由云平台处理后推送至组态页面,平台接收边缘网关采用 MQTT 协议上报的灯具实时数据并通过 WebSocket 协议推送数据的结果如图 17 所示。

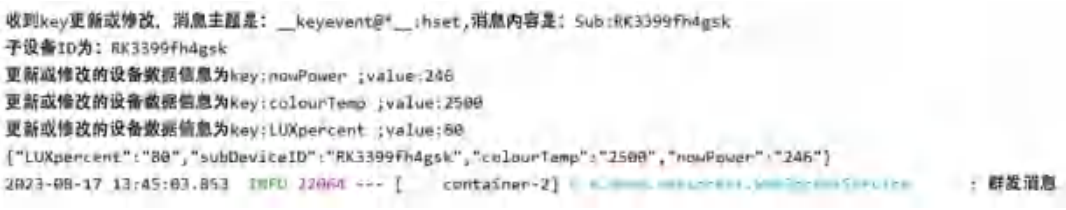

图 17 MQTT 消息接收与数据推送结果

Fig. 17 MQTT message reception and data push results

3.4 历史参数统计功能

利用云组态平台数据处理量大、稳定性好的优点,可对智能家居系统中的能耗情况、故障情况和当前系统状态进行实时统计,2022 年 9 月 10 日至 2022 年 9 月 21 日的系统能耗情况、在这段时间内边缘网关下子系统中的故障信息,以及目前系统的电压、电流和功率情况如图 18 所示。

图 18 历史数据统计页面

Fig. 18 Historical data statistics page

4 结束语

本文设计并实现了一种基于云平台的组态式智能家居人机交互系统,以图形组态的方式搭建智能家居场景,操作更加简便、直观;本文还设计了系统数据库存储模型和数据更新方法,相比与传统的定时轮询的数据更新方式提高了数据实时性;最后,对该平台进行了相应的测试,实验结果表明该系统可正常实现场景搭建、数据更新和历史数据统计等各项功能,为智能家居人机交互系统的开发和管理提供了一种更便捷、高效的解决方案。

参考文献

[1] 肖丁, 王乾宇, 蔡铭, 等. 智能家居场景联动中基于知识图谱的隐式冲突检测方法研究[J]. 计算机学报, 2019, 42(6): 1190-1204.

[2] 盖森, 鲁艺, 张印帅,等. 面向智能家居系统的主动交互设计研究[J]. 计算机辅助设计与图形学学报, 2023, 35(2): 230-237.

[3] LAZAREVIC I, SEKULIC M, SAVIC M S, et al. Modular home automation software with uniform cross component interaction based on services[C]//Proceedings of 2015 IEEE 5th International Conference on Consumer Electronics. Piscataway, NJ: IEEE, 2015:363-365.

[4] 胡波，李响，陈宏君，等. 全国产分散控制系统组态软件框架的设计和实现[J]. 热力发电，2021，50(12)：13-20.
[5] 张成，李迪，吴培浩，等. 基于 Web 发布的组态软件设计与实现[J]. 自动化与仪表，2018，33(1)：89-92.
[6] 张世荣，童博. 基于云的可组态 PLC 实验评判系统设计[J]. 实验室研究与探索，2020，39(8)：65-69.
[7] 陈清. 基于 Redis 的煤矿大型机电设备联网架构[J]. 工矿自动化，2020，46(10)：109-113.
[8] SAYAR A，ARSLAN Ş，ÇAKAR T，et al. High-performance real-time data processing：Managing data using debezium，postgres，Kafka，and Redis[C]// Proceedings of the 2023 Innovations in Intelligent Systems and Applications Conference (ASYU). Piscataway，NJ：IEEE，2023：1-4.
[9] 何鹏飞，何平，张琼琼，等. 支持 ModBus 协议的组态式人机界面系统的设计[J]. 组合机床与自动化加工技术，2015(5)：38-42.
[10] MENDES E J，SILVEIRA M M，ARAÚJO M B，et al. Abstraction of heterogeneous iot devices for management of smart homes[C]// Proceedings of 2019 IEEE Latin-American Conference on Communications (LATINCOM). Piscataway，NJ：IEEE，2019：1-6.
[11] 马腾霄. 数控系统 HMI 组态化技术研究[D]. 武汉：华中科技大学，2017.
[12] FAZLOLLAHTABAR H. Internet of Things-based SCADA system for configuring/reconfiguring an autonomous assembly process[J]. Robotica，2022，40(3)：672-689.
[13] PAPP I，VELIKIC G，LUKAC N，et al. Uniform representation and control of Bluetooth Low Energy devices in home automation software[C]// Proceedings of 2015 IEEE 5th International Conference on Consumer Electronics. Piscataway，NJ：IEEE，2015：366-368.
[14] KÖLSCH J，POST S，ZIVKOVIC C，et al. Model-based development of smart home scenarios for IoT simulation[C]// Proceedings of the 8th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems. Piscataway，NJ：IEEE，2020：1-6.
[15] LCHEV S，OTSETOVA-DUDIN E. Device model and communication protocol with low overhead for sensors and actuators in smart buildings[C]//Proceedings of the 23rd International Conference on Computer Systems and Technologies. Piscataway，NJ：IEEE，2022：33-38.
[16] 王培屹，张桂青，李清锋，等. 农机制造物联网实时数据采集及处理系统[J]. 制造业自动化，2019，41(11)：23-27.