

基于云边协同的电动汽车共享充电控制系统设计与实现

向宁¹, 杨琴¹, 曾振柄², 陈斌³

1. 乐山职业技术学院 人工智能学院, 四川 乐山 614000

2. 上海大学 理学院, 上海 200444

3. 四川百威智联科技有限公司, 四川 乐山 614000

DOI: 10.61369/TACS.2025070012

摘 要 : 随着5G+AIoT技术的深度应用, 基于MQTT协议的远程控制技术成为工业数字化转型的关键支撑。本文面向老旧小区电动汽车共享充电场景, 设计了一种轻量级的远程控制系统, 依托MQTT协议的低带宽、高效率特性, 通过ESP32硬件平台与Lua语言实现设备的实时管控。系统采用发布/订阅模式构建了三层架构: 边缘层完成充电桩数据采集与指令执行, 云端层部署MQTT代理服务器实现消息路由, 应用层提供远程监控与策略管理。研究内容涵盖系统需求分析、总体架构设计、核心业务逻辑、硬件设计、充电小程序设计等方面。研究成果为老旧小区实施共享充电设施规模化部署提供了高可靠、可扩展的解决方案, 助力推动数字社区与智能交通协同发展。

关 键 词 : 云边协同; MQTT 协议; 共享充电; 智能调度

Design and Implementation of an Electric Vehicle Shared Charging Control System Based on Cloud-Edge Collaboration

Xiang Ning¹, Yang Qin¹, Zeng Zhenbing², Chen Bin³

1. School of Artificial Intelligence, Leshan Vocational and Technical College, Leshan, Sichuan 614000

2. School of Science, Shanghai University, Shanghai 200444

3. Sichuan Baiwei Zhilian Technology Co., Ltd., Leshan, Sichuan 614000

Abstract : With the deepened application of 5G and AIoT technologies, remote control systems based on the MQTT protocol have emerged as a critical enabler for industrial digital transformation. This paper focuses on the scenario of shared electric vehicle charging in aging residential communities, proposing a lightweight remote control system. Capitalizing on the low-bandwidth and high-efficiency features of the MQTT protocol, real-time device management is achieved through an ESP32 hardware platform combined with Lua programming. The system adopts a three-layer architecture built upon a publish/subscribe model: the edge layer handles data collection from charging piles and executes commands; the cloud layer deploys an MQTT broker server for message routing; and the application layer offers remote monitoring and policy management. Research contents encompass system requirement analysis, overall architectural design, core business logic, hardware design, and charging mini-program design. The research findings provide a highly reliable and scalable solution for the large-scale deployment of shared charging facilities in old residential areas, facilitating the coordinated development of digital communities and intelligent transportation.

Keywords : cloud-edge collaboration; MQTT protocol; shared charging; intelligent scheduling

引言

近年来, 购买新能源电车的家庭越来越多, 充电难的问题越来越凸显。针对当前老旧小区, 受电力容量及布线条件限制等因素的影响, 安装充电桩的数量受到限制, 不能保证每个电车用户均能申请到充电桩的安装许可。调研发现, 当前充电桩使用存在以下主要问题: 一是充电桩综合利用率低。据国家能源局统计, 我国公共充电桩综合利用率低于15%, 充电桩运营商收回投资较为困难, 利用率低

项目信息: 乐山市科技局高新科技领域重点研究项目, 立项编号: 23GZD009, 项目名称: 住宅小区电动汽车共享充电控制系统设计。

作者简介: 向宁(1978—), 男, 四川遂宁人, 硕士, 副教授, 主要研究方向为人工智能、软件开发。E-mail: 550666095@qq.com

下导致许多充电桩处于无人维护、甚至停业的状态；二是居民充电桩闲置率高。电车作为代步车辆，往往不需要每日充电，根据车辆行驶每天平均50公里交通，电车平均满电续航360公里以上，余20%左右充电，则6天充一次电，而其余5天则该充电桩一直处于闲置状态；三是居民小区充电桩申请难。老旧小区申请安装充电桩受本区域电力容量的限制，许多小区现有电能除了供居民家庭生活及物业公共设施供电外，所剩无几，因此对于一个平均1千户的居民小区，可供安装的充电桩容量往往不超过50个。

一、系统总体架构设计

系统由云端层、边缘层和终端层所组成。其中（1）云端层的Web服务器负责处理来自微信小程序和边缘设备的请求，提供数据存储、业务逻辑处理、用户认证与授权等服务，能够有效整合系统的核心功能，确保系统的高效运行和可扩展性^[1]。云端层的MQTT服务，用于负责处理来自边缘设备的实时数据交换，如充电状态、电量信息等；MQTT协议因其低开销、低带宽占用和实时性强的特点，非常适合用于物联网场景的数据通信。数据库选用MS SQL，用于存储用户信息、充电设备信息、充电记录、支付信息等关键数据^[2]。（2）边缘层的智能充电设备（UE端）具备充电、计量、保护等功能，并能够通过MQTT协议与云端服务器进行通信，用以上传当前充电电压、充电电流、充电电量等数据到云端服务器，并接收来自云端的控制指令。（3）终端层的微信小程序，作为用户与系统进行交互的主要界面，提供了充电预约、充电状态、充电支付、充电记录查看等功能。系统总体架构如图1所示。

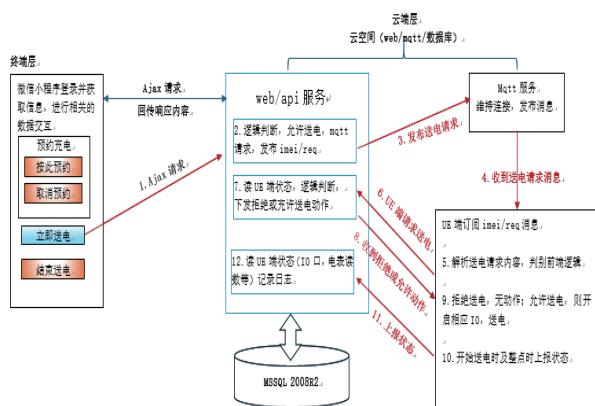


图1 系统总体架构图

二、系统核心业务逻辑

用户通过小程序发起预约或即时充电请求，请求经HTTPS加密上传至云端。云端调度中心进行身份验证、排队决策并应用智能调度算法，生成控制指令。指令通过MQTT协议下发至边缘设备，驱动接触器执行充电启停及数据采集。边缘设备实时采集的充电数据再次通过MQTT上报至云端，触发状态更新与自动计费。最终系统生成账单并推送至用户端完成支付，形成业务闭环。该系统以云端调度为核心决策层，依托MQTT实现云边高效、可靠的双向通信。系统核心业务逻辑流程如图2所示。

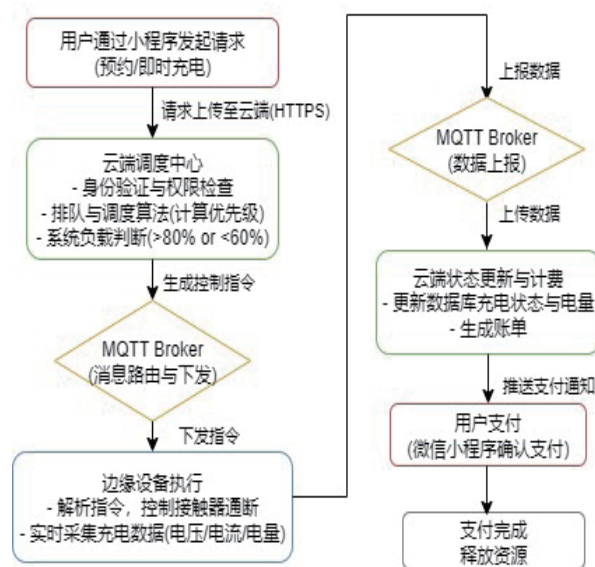
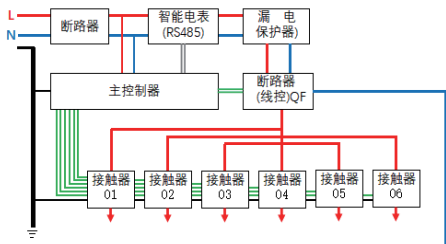


图2 系统核心业务逻辑流程图

三、系统硬件设计

（一）共享充电控制设备硬件设计

在共享充电桩的设计方案中，通过在单个充电桩电表上加装特定的共享充电控制装置，实现6个充电桩的共享充电功能。这些充电桩并非同时处于充电状态，在任何时刻，最多仅允许一个充电桩进行送电操作。本方案采用“5+1”的共享模式，针对7KW与4KW的充电桩组合提出相应的解决方案。具体而言，对于一个核定电功率为11KW的充电桩电表，它能够满足7KW和4KW充电桩同时送电的需求，即为一个7KW充电桩和一个4KW充电桩提供电力支持。鉴于目前大多数家用混动车的充电桩功率为4KW，采用这种模式可以显著提升充电桩的共享性和充电效率^[4]。共享充电控制设备内部结构示意图如图3所示，主要由断路器、智能电表等组成。其中断路器用于控制柜内总控电源的输入控制，控制整个电路的送电，充电电表来电接断路器上端；智能电表用于记载功率及电量，与控制器通过485通讯线连接；漏电保护器用于控制对外共享充电桩的总电源；断路器（线控）QF由主控制器控制通断，通电后用于对接触器提供相电供；主控制器用于读取智能电表实时电量读取，将相关的电表电量读数及接触器状态上报云端，依据云端返回指令进行相关的操作，如控制接触器通断；接触器：其控制线圈为DC12V由主控制器依据内部逻辑提供，当控制线圈通电时，则吸合触点，连通^[5]。



开机后主控制器自检，所有 IO 口均处于低电平，线控断路器开路，接触器开路；主控制器入网后，向云平台上报开机信息。平台依据后台逻辑，下发控制命令，主控制器读取云平台下发的控制命令，进行命令解析，进行相应的逻辑处理^[6]。连接 MQTT 服务器，定时发布设备状态信息，如设备 IMEI 码、设备 ID、接触器状态、电表总电量读数、时间戳、数据签名等。订阅 MQTT 消息，收到 MQTT 消息签名验证，签名验证后，判别签名处理，支持立即读 MQTT 消息、上报 MQTT 状态^[7]。读 MQTT 消息，则立即向 MQTT 消息发布设备状态信息；上报 MQTT 消息，则向云平台上报设备信息。控制设备软件运行流程如图 4 所示。