密级：公开

高压杆塔螺栓自动巡检系统的

设计与实现

**Design and Implementation of Automatic Inspection System for Bolts of High Pressure Tower**

|  |  |
| --- | --- |
| **学 院：** | 软件学院 |
| **学 号：** | 151203411 |
| **专业班级：** | 软件工程 1504班 |
| **学生姓名：** | 郑雪峰 |
| **指导教师：** | 邵中 |

2019 年 6 月

摘 要

随着物联网技术的发展，电力系统智能化也得到了迅猛发展，杆塔巡检作为电力系统中的重要组成部分，急需完成智能化转型。利用物联网技术实现传感器与云端服务的互联，实时监测部署，统一指挥，具有重要实际意义。

为优化电力系统杆塔巡检工作流程，基于“泛在电力物联网”的思想和辽宁省电力有限公司的实际需求，开发了可以自动定期检测高压杆塔上螺母是否松动并针对异常终端进行报警、定位引导、排查工单安排的杆塔巡检系统。

本文以边缘网关组态控制程序、数据可视化平台、业务逻辑服务程序三个核心子系统为设计重点。边缘网关组态控制程序采用Serial端口与Socket技术连接AD数据采集终端与数据服务器，并根据数据类别、数据等级加密后分发至不同服务器进一步处理，以实现“边缘计算”的功能。数据可视化平台采用B/S架构，使用Vue.js框架进行数据绑定，使用Bootstrap框架进行响应式布局，可适用于不同尺寸的展示大屏幕；采用OCX控件连接监控设备并将画面实时展示在可视化平台中；采用腾讯地图API在地图中标注杆塔位置，并利用Canvas蒙层以非均匀有理B样条曲线描绘杆塔闭合区域，解决了在屏幕墙输出时杆塔区域边界锯齿过于明显的问题。业务逻辑服务程序基于Spring Boot框架开发，实现前端与后台完全分离，提供统一的数据接口API，方便二次开发。

系统实现了AD采集终端数据的边缘处理、采集终端的系统组态功能、数据可视化平台的实时数据展示、实时报警、工单调度功能。经使用单位测试表明，系统功能完整，运行稳定，能够满足杆塔螺栓自动巡检的需要，对解决高压杆塔维修难、定位难、人力调动难、巡检难等问题具有实际价值，对同类产品的研发具有一定参考价值。

**关键词：**智能螺栓；杆塔巡检；泛在电力物联网；Socket；Spring Boot

**Abstract**

With the development of the Internet of Things technology, the intelligentization of the power system has also developed rapidly. As an important part of the power system, the poles and towers inspection is urgently needed to complete the intelligent transformation, using the Internet of Things technology to realize the interconnection of sensors and cloud services, real-time monitoring and deployment. Unified command has important practical significance.

In order to optimize the power system tower inspection work process, according to the actual needs of the State Grid Liaoning Electric Power Company, based on the idea of “Ubiquitous power Internet of things”, to solve the difficulty of poles and towers inspection, a set of automatic high-voltage towers can be detected on a regular basis. Whether the nut is loose, and sends out an alarm message, positioning and guiding maintenance personnel to investigate and unify the deployment of the tower inspection system for disaster relief ,to better solve the difficult problems of high-voltage tower maintenance, positioning, manpower transfer, inspection and so on. The edge gateway configuration control program, data visualization platform, and business logic service program are the core subsystems of the system. For the edge gateway configuration control program, the system uses the Serial port and Socket technology to connect the AD data collection terminal and the data server, and encrypts it according to the data category and data level and distributes it to different servers for further processing to realize the function of “edge calculation”; For the data visualization platform, the system adopts the B/S architecture, uses the Vue.js framework to bind the data to the visualization platform, and uses the bootstrap framework for layout, ensuring that the visualization platform is suitable for most large screens, and the OCX controls are used to connect the monitoring devices. Real-time display in the visualization platform, the system uses the API provided by Tencent map to realize the position of the tower, and uses the non-uniform rational B-spline to realize the closed area drawing display. The closed curve generated by Canvas is drawn in the mask through the map API; For the business logic service, the business logic service is deployed in the data server and the business server, and the Spring Boot framework is used to write the API, which realizes complete separation of the front-end background and unified data interface format, which is convenient for secondary development.

The system realizes the edge calculation of the edge gateway, the system configuration of the acquisition terminal, the real-time data display of the display platform, the real-time alarm, and the unified command. The test unit test shows that the system function is relatively complete and can meet the needs of the automatic inspection of the pole bolts. It is stable in operation and has certain reference value for the operation mechanism and research and development of similar products.

**Keywords:** Intelligent bolt; Tower Inspection; Ubiquitous power Internet of Things; Socket; Spring Boot

目 录

[摘 要 I](#_Toc11588568)

[Abstract II](#_Toc11588569)

[第 1 章 引 言 1](#_Toc11588570)

[1.1 课题研究的目的与意义 1](#_Toc11588571)

[1.2 研究现状 1](#_Toc11588572)

[1.3 设计内容 2](#_Toc11588573)

[1.4 实验环境 3](#_Toc11588574)

[第 2 章 相关理论及技术分析 4](#_Toc11588575)

[2.1 理论分析 4](#_Toc11588576)

[2.1.1 线性同余法 4](#_Toc11588577)

[2.1.2 AES加密算法 5](#_Toc11588578)

[2.1.3 缓冲中断处理机制 5](#_Toc11588579)

[2.1.4 非均匀有理B样条 6](#_Toc11588580)

[2.2 技术分析 7](#_Toc11588581)

[2.2.1 Vue.js框架 7](#_Toc11588582)

[2.2.2 Ajax技术 8](#_Toc11588583)

[2.2.3 Socket通讯技术 9](#_Toc11588584)

[第 3 章 系统需求分析与总体设计 10](#_Toc11588585)

[3.1 业务模式分析 10](#_Toc11588586)

[3.2 软件需求分析 10](#_Toc11588587)

[3.2.1 边缘网关组态程序需求定义 11](#_Toc11588588)

[3.2.2 业务服务需求定义 12](#_Toc11588589)

[3.2.3 数据展示平台需求定义 13](#_Toc11588590)

[3.4 系统总体设计 13](#_Toc11588591)

[3.4.1 数据可视化展示平台设计 14](#_Toc11588593)

[3.4.2 业务服务程序设计 14](#_Toc11588594)

[3.4.3 边缘网关组态程序设计 15](#_Toc11588595)

[3.5 数据库设计 16](#_Toc11588596)

[3.5.1 System\_Mian数据库设计 16](#_Toc11588597)

[3.5.1 SSO数据库设计 20](#_Toc11588598)

[第 4 章 关键算法设计 24](#_Toc11588599)

[4.1 基于序列偏移化的设备合法性验证算法 24](#_Toc11588600)

[4.1.1 随机数生成算法 25](#_Toc11588601)

[4.1.2 序列偏移化算法 26](#_Toc11588602)

[4.2 实时信息流推送权限管理 27](#_Toc11588603)

[4.3 杆塔连线渲染算法 28](#_Toc11588604)

[4.4 高并发信号缓冲机制 30](#_Toc11588605)

[4.5 基于非均匀有理B样条的闭合区域绘制 33](#_Toc11588606)

[第 5 章 系统功能的设计与实现 36](#_Toc11588607)

[5.1 边缘网关组态程序 36](#_Toc11588608)

[5.1.1 通信参数配置模块 36](#_Toc11588609)

[5.1.2 终端操作排查模块 38](#_Toc11588610)

[5.1.3 信息配置模块 38](#_Toc11588611)

[5.1.4 系统组态 39](#_Toc11588612)

[5.2 数据展示平台 39](#_Toc11588613)

[5.2.1 单点登录 40](#_Toc11588614)

[5.2.2 报警提示模块 41](#_Toc11588615)

[5.2.3 抢修人才、资源展示模块 43](#_Toc11588616)

[5.2.4 杆塔区域展示模块 44](#_Toc11588617)

[第 6 章 结 论 48](#_Toc11588618)

[在学研究成果 49](#_Toc11588619)

[参考文献 50](#_Toc11588620)

[致 谢 51](#_Toc11588621)

第 1 章  引 言

## 1.1 课题研究的目的与意义

在科技高速发展的今天，物联网技术已经慢慢渗透进了人们的生活，物联网将现实世界[数字化](https://baike.baidu.com/item/%E6%95%B0%E5%AD%97%E5%8C%96)，统一物与物的数字信息并且已经在运输和物流领域、工业制造等领域已产生了不俗的成果。2019年，国家电网有限公司1号文件中提出“泛在电力物联网”概念[1]，使用统一接口利用统一边缘路由的方式实现智能电网的管理、数据交换等功能，人工电网杆塔巡检完全可以使用价格低廉的电子感应装置以及配套的系统所替代，普通输电线路一般采用架空线路，需承受来自重力、风力、暴雨、雷电和冰雪等机械力的作用与风沙等的侵蚀[2],运行环境有时较为恶劣。当输电线路经过沙漠地带、高盐土质区、采空区和山地滑坡区等不良地质区，在自然环境和外界条件的作用下，杆塔容易出现脱锌、掉漆、螺栓松动甚至损坏等情况而导致供电线路陷于瘫痪，造成巨大损失。目前，国内主要依靠人工的方式检修杆塔，耗费大量人力物力，同时线路巡检时存在巡检记录无统一规范、巡检人员巡检时难以考勤、统计人员错登杆时无法提醒、靠近危险点时无法提示等问题[3]。

基于“泛在电力物联网”的思想，本课题以解决杆塔巡检难为目的，构建了一套可以自动定期检测高压杆塔上螺栓是否松动，并发出报警信息，定位引导维修人员排查、统一调动部署救灾的杆塔巡检系统，对解决高压杆塔维修难、定位难、人力调动难、巡检难问题有重要意义。

## 1.2 研究现状

最早提出“智能电网巡检”的是德国2009年的“E—Energy”计划，德国西门子、SAP及瑞士ABB等企业均参与了这一计划[4]，并在之后的4年在全国6个试点城市取得了非常好的成效。目前世界上许多国家也在进行智能电网巡检系统的研发工作，“边缘网关数据处理”、“高电压下的信号稳定性”等问题为该方向的热点研究问题，光纤技术、ZigBee通讯协议、神经运算等传统IT行业解决方案已经尝试在这一领域发挥作用，并已经取得了成效。

目前针对杆塔巡检难问题，国内一些研发机构、学校、企业等已经通过研究实践在一定程度上实现了同类产品的应用。这些产品的检测方式主要为被动式检测、图像检测等。

基于压电阻抗技术的螺栓松动监测系统是同济大学结构工程与防灾研究所与日本东北工业大学工学部共同开发，其工作主站作为信号接收端，通过烧写在主站控制主板中的控制程序向从站轮询当前螺栓压力信息[5]并上传至数据中心，该系统能够实时检测螺栓的压力值，并在检测中心显示，但该系统适用于采集端较少的生产环境，智能电网中随着杆塔螺栓的数量增加，该检测方案上传的大量数据会对数据中心的服务器造成过大压力，对资源消耗过大。

图像识别智能螺栓拧紧检测系统是西南交通大学机械工程学院研发的一套螺栓检测系统，系统通过扭矩控制的原理和图像识别的方法检测螺栓是否松动[6]，该方法适用于铁路、道路等静止物体上的检测，而输电杆塔的绝缘子时常会因为风、雪等原因发生偏移，并且摄像头采集的图像会因为杆塔产生的电弧而产生大量噪声点，影响系统判断。

2011年，韩国建筑工程学院提出了主动式螺栓松动检测系统，该系统用于检测韩国国内桥梁中螺栓连接处螺栓松动情况，无需检测人员定期检测，螺栓松动后触发ZigBee射频模块并发出信号，工程人员接收相应信号后可直接检查维护相应区域，同时，该系统通过无线控制的方式对螺栓进行紧固操作，并实时检测螺栓紧固器的工作情况[7] ，但上述系统均存在造价高昂、功耗大等问题。

目前，全国范围内仅江苏供电公司实现了部分杆塔巡检系统的应用，但也仅限于电流、电压检测等[8]，其使用的电流传感器会对无线传输模块造成相应干扰[9]并且由于杆塔周围高压电场的影响，普通信号传输方式会受到较大影响导致上述系统无法直接应用在高压杆塔上，部分供电公司巡检采用无人机巡检的方式，但无人机平台受环境影响较大，司空较为复杂，故无法广泛使用。

2017年河南省电力公司发明出了一套“内置式光纤智能螺栓检测系统”，该系统通过光纤进行信号传输，能够较大程度上屏蔽高压电弧带来的干扰问题，该系统目前尚未进行测试使用。

## 1.3 设计内容

本系统由AD采集终端(智能螺栓)、边缘网关、业务服务组成，当AD采集终端部署在对应杆塔后，终端开始接收终端控制器的控制信号，终端控制器通过定时轮询向终端发送数据接收请求，AD采集终端采集压力信号后，通过Serial端口将数据发送至边缘网关；边缘网关可以进行数据处理、数据计算等操作，边缘网关与业务服务器采用Socket方式连接，普通数据直接由边缘网关处理后存入对应数据库中，报警、控制信号等突发数据通过WebSocket与数据可视化平台直接连接；数据可视化平台通过WebSocket与业务服务器进行长连接，报警数据经服务器路由转发后最终将展示在数据展示平台中。

## 1.4 实验环境

本课题智能螺栓部分采用STM32、CC2530与FSR压电薄膜组成的AD采集终端作为智能螺栓的核心元器件。边缘网关组态程序采用.Net Framework 4.7框架的Windows Form编写，其简洁的控件形式并以控件为基本单位的模块化方案，对本系统组态程序的开发与后期工作人员的使用提供了极大的方便。本系统使用Tomcat服务器部署Spring Boot编写的API，服务器与数据展示平台采用API的形式进行数据传输，对后期的服务变更、优化开发提供了便利。

第 2 章  相关理论及技术分析

本系统中，采用线性同余法生成均匀随机数作为参数，利用非均匀有理B样条进行地域曲线的绘制，利用AES加密算法对控制信号进行对称加密，利用环形缓冲队列对摄像机传输的视频数据信号进行缓冲，边缘网关利用Socket通讯与服务器连接，数据展示平台利用WebSocket长连接与服务器进行连接，数据展示平台利用Vue.js框架进行双向数据绑定，边缘网关利用Windows Form框架OCX控件对终端设备驱动、联动方式进行组态。

## 2.1 理论分析

本系统中采用线性同余法生成随机数，通过AES加密算法对数据进行加密，采用环形缓冲队列对数据进行缓冲罐，采用NURBS曲线实现闭合区域绘制。

### 2.1.1 线性同余法

线性同余法也称“线性同余随机数生成器”、LCG法。主要以数论中的同余算法生成随机数，是产生[0，1]均匀分布随机数的方法之一，包括混合同余法和乘同余法[10] 。初等数论中，若两数a、b模m取余，所得余数相同，则称a模m同余b，如式(2-1)。

(2-1)

将一侧改写为线性函数，可得线性同余公式如式(2-2)所示。

(2-2)

式(2-2)中，X为自变量，Y为因变量，X与参数A相乘后与参数B相加，最后对参数M取余，所得Y即为一个相对X随机的随机数。所需随机数为非负整数时，自变量X及参数A、B、M均为正整数。为提高随机性，A与B需取较大的值，A取，B取,且A与M互质可获得较好的随机性。

通过改变自变量X的取值可得到连续的随机数。为避免使用线性变化的自变量获取随机数时所得随机数列出现线性关系，采用迭代法，将上一个因变量的取值作为下一个自变量的取值，公式如式(2-3)所示。

(2-3)

为所得随机数列，称为随机种子，取任意非负整数即可。此时自变量数列为随机数列，可增强所得数列的随机性。采用该方法获得的随机数列在取适当参数时能通过随机数检验，具有较好的随机性。本系统采用该方法产生随机数序列用于加密算法的加密参数。

### 2.1.2 AES加密算法

AES算法是对称密钥加密方式之一，根据密钥长度不同，其加密方式加密轮的次数也不同，本课题采用了AES-128的加密模式，加密轮数为10轮。由于AES加密算法为对称加密，故本算法分为加密函数与解密函数，加密函数和解密函数共同使用同一个密钥K，当明文字符串P需要进行传输时，加密方使用加密函数E和加密密钥K可以获得对应密文C即，并将密文C传递至解密方，解密方接收密文后，使用解密函数D和密钥K以获取明文内容，即[11]。

使用AES加密算法进行加密时，需要进行字节替换、行位移、列混淆与轮密钥加四步操作，解密算法为此四种操作的逆操作。

字节替换，加密函数中通过S盒实现由一个字节到另一个字节的映射，解密函数中，使用逆S盒完成上述映射，S盒与逆S盒均为16阶方阵，通过S盒替换可以增加混淆性。

行位移，行位移操作是一个4阶状态方阵中字节间的置换操作，用于向算法提供扩散性，在AES-128模式加密函数中行位移操作会将状态矩阵的第k行左移k个字节；解密函数中进行相反的位移操作。

列混淆，列混淆利用域上算数特性的一个替代，用于向算法提供扩散性。列混淆通过矩阵间的相乘来实现。经过行位移变换后的状态阵与固定阵相乘即可得到混淆后的状态矩阵。加密函数中使用的固定矩阵与解密函数中使用的固定矩阵互为逆矩阵。

轮密钥加，该操作是将128位密钥与最后得到的状态矩阵中的数据进行逐位异或操作。因AES-128模式需要进行10轮加密，在加密过程中需要对初始密钥进行密钥拓展操作。每一轮密钥拓展操作根据上一轮密钥拓展操作得到的子密钥进行重复的操作。

对称加密算法中AES算法具备极高的安全性，与非对称算法相比，AES算法加密速度快，灵活性强，只需要少量的储存器。

### 2.1.3 缓冲中断处理机制

因串口数据处理机制为接收数据后，进入中断并在中断函数中将数据读取出来进行处理。这一种数据处理机制是“非缓冲中断方式”，这种数据处理方式无缓冲区，如果先前接收的数据尚未处理完成，又接收到新的数据，新接收的数据就会把尚未处理的数据覆盖，从而导致“数据丢包”[12]。此时需要建立相应的缓冲区对数据进行缓冲，普通队列缓冲区在用掉一个数据元素后，其余数据元素需要向前搬移而环形缓冲队列不需要移动存储位置，也不需要进行动态的内存释放和分配，使用固定大小的内存空间反复使用。在实际的队列插入和弹出操作中，是不断交叉进行的[13]，因而可以作为系统缓冲方案的首选，环形缓冲队列如图2-1所示。

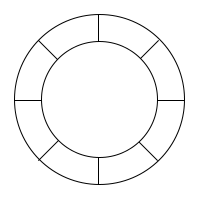


图2-1 环形缓冲队列

### 2.1.4 非均匀有理B样条

非均匀有理B样条曲线是在贝塞尔曲线的基础上发展的一类曲线，是贝塞尔曲线的一般化。B 样条曲线根据节点分布分为均匀B 样条曲线、准均匀B 样条曲线和非均匀B 样条曲线。本课题使用的是非均匀B 样条曲线算法。

非均匀B 样条曲线算法实现主要分为三步：即根据提供的控制点集计算出节点向量；根据计算出的节点向量计算出k次B 样条基函数；根据控制点和计算出的基函数计算出曲线的任意点。

由n+1个控制点构成了控制点集（i = 0,1,2,……, n），按顺序连接控制点可构成一个封闭的多边形。依次将控制点代入曲线方程可得到曲线上的点。K阶B-样条曲线的表达式如式(2-4)所示。

(2-4)

阶B-样条基函数采用deBoor和Cox给出的递归定义，公式如式(2-5)所示。

(2-5)

其中规定

其中*(i = 0,1,2,……,n)*是根据控制点计算出来的节点集。

对于闭合曲线，计算节点集时需要根据B-y样条曲线阶数K将控制点重叠使用。根据控制点集计算节点集有多种计算方法，本课题采用哈德利-贾德法，计算公式如式(2-6)所示。

(2-6)

其中为点和点的距离。

## 2.2 技术分析

本系统中前端框架采用Vue.js框架，通过Ajax进行局部刷新，采用Socket进行服务器与边缘网关的通讯。

### 2.2.1 Vue.js框架

  Vue.js是一个轻巧、高性能的渐进式框架，仅专注于视图层，其数据的双向绑定减少了DOM操作，并且Vue 可以驱动单一文件组件，使用API 实现响应的数据绑定和组合的视图组件，适用于构建本课题所需的数据驱动的Web应用界面[14]。 Vue利用“Object.defineProperty()”方法实现双向数据绑定，该方法重新定义了获取属性值get与设置属性值set，接收三个参数，分别为操作的对象，定义或修改的对象属性与属性描述符。

实现MVVM的双向数据绑定采用数据劫持与发布者-订阅者的模式，通过“Object.defineProperty()”劫持属性的setter与getter，在数据发生变化时将消息发布到订阅者，触发相应的监听回调，协作方式如下图2-2所示。



图2-2 MVVM协作方式

实现数据监听器Observer可以对数据对象的属性进行监听，数据变动可以取得最新值并通知订阅者，指令解析器Compile对每个元素节点的指令进行扫描、解析，根据指令模板替换数据及绑定相应的更新，Watcher监听器连接Observer与Compile，可以订阅收到属性变动，执行回调函数，更新视图。

### 2.2.2 Ajax技术

Ajax是一种无需重新加载整个网页即可进行局部刷新的技术，本系统数据展示平台数据来自于多个异步变化的数据源，数据展示平台需要创建XMLHttpRequest对象并创建新的Http请求，指定请求的URL及验证信息后发送Http请求至后台并创建HttpResponse对象以接收返回数据，服务器通过HttpResponse对象返回数据，数据展示平台获取异步调用返回的数据实现页面的局部刷新，时序图如图2-3所示。



图2-3 Ajax 工作过程时序图

### 2.2.3 Socket通讯技术

Socket用于描述IP地址和端口，是一个通信链的句柄，可以用来实现不同虚拟机或不同计算机之间的通信。在Internet上的[主机](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA/455151)一般运行了多个服务软件，同时提供几种服务。每种服务都打开一个Socket，并绑定到一个端口上，不同的端口对应于不同的服务。本课题边缘网关驱动程序与业务逻辑服务程序之间采用Socket通讯的方式传输数据，数据展示平台与业务服务之间采用WebSocket方式进行数据传输。使用Socket的三次握手连接的特性可以有效的提高系统数据传输的可靠性[15]，Socket的三次握手协议时序图如图2-4所示。



图2-4 Socket三次握手时序图

系统建立连接时，客户端发送同步序列编号syn=j，并进入SYN\_SENT状态，服务器收到同步序列编号syn=j，需要确认客户端的同步序列编号ack=j+1，同时发送服务器自己的同步序列编号seq=k，此时服务器进入SYN\_RECV状态，客户端接收到服务器的同步序列编号，向服务器发送确认包ack=k+1，客户端和服务器TCP连接成功，完成三次握手。

第 3 章  系统需求分析与总体设计

本系统构建了一套可以自动定期检测高压杆塔上螺母是否松动，并发出报警信息，定位引导维修人员排查、统一调动部署救灾的杆塔巡检系统。

## 3.1 业务模式分析

传统杆塔巡检中首先由巡检规划人员对杆塔巡检规划并生成《巡检规划书》，经过安全人员审核后将巡检计划存档并将《巡检规划书》发放到巡检指挥人员、对应变电站工作人员以及巡检人员处，巡检人员根据《巡检规划书》按时对需要巡检的杆塔进行巡检，在巡检过程中，正常杆塔需要巡检人员出具《巡检详情表》并交由变电站工作人员生成巡检记录存档；对于需要维修的杆塔，巡检人员需要出具《维修申请表》并进行维修预约，维修人员接收预约申请后在预约时间对指定杆塔维修，维修完毕后需要出具《维修记录表》并存档，当重大险情发生或在巡检过程中出现不适宜预约维修的险情，需要巡检人员出具险情登记表并直接拍照上传至指挥人员处，指挥人员直接通过电话指挥对应值班维修人员对杆塔进行维修，业务流程图如图3-1所示。



图3-1 杆塔巡检业务流程图

## 3.2 软件需求分析

在传统人工巡检、维修流程中，存在大量单据流转，实时性较差等问题，在维修预约等待过程中，待维修的杆塔可能已经出现险情，造成不可挽回的损失。在杆塔出现重大险情时，通过电话通知的方式很可能会出现因指挥人员表述不清而导致抢险遗漏等情况的发生，在维修人员寻找问题杆塔时，也会出现因不熟悉杆塔位置而找不到对应杆塔而贻误险情的问题。

### 3.2.1 边缘网关组态程序需求定义

边缘网关组态程序应部署在杆塔下方的主机中，应实现通信参数配置功能，终端操作功能、终端关闭、重启功能、数据库配置功能、POST配置功能功能、Socket配置功能。系统用例图如下图3-2所示。



图3-2 边缘网关组态程序用例图

系统角色分为杆塔巡检人员与系统管理员，其中数据库配置、POST配置、Socket配置应由巡检管理员进行管理，其余功能应由杆塔巡检人员进行管理；系统应采用图形可视化界面与用户进行交互。

通讯参数配置应提供负责数据通信的串口配置功能以及负责报警推送的串口配置功能；终端操作应提供终端设备的添加、删除功能；关闭、重启终端应提供AD采集设备的单一设备关闭、重启以及杆塔中所有设备的关闭、重启功能；数据库配置中应提供数据库连接配置功能；POST配置中应提供POST服务器配置以及服务接口配置功能；Socket配置中应提供边缘网关与对应业务服务器的Socket套接字信息的配置功能。

### 3.2.2 业务服务需求定义

业务服务应部署在数据中心的业务服务器中，系统用例图如下图3-3所示。



图3-3 业务服务程序用例图

系统应向连接到服务器的边缘网关、数据可视化展示平台提供Socket管理功能，包括Socket的添加与删除功能；同时，系统应面向边缘网关提供终端设备服务功能，包括终端设备添加指令与删除指令的转发功能；系统应向系统管理员提供数据管理功能以管理报警信息或普通数据信息，系统应向数据可视化展示平台提供报警推送功能，通过Socket服务进行报警推送。

### 3.2.3 数据展示平台需求定义

数据展示平台应为部署在各电网属地分公司服务器中的Web应用程序，以数据展示，实时报警，终端控制，统一部署调动指挥为目的的数据展示大屏。用例图如下图3-4所示。



图3-4 数据可视化展示平台程序用例图

数据可视化展示平台应提供报警信息展示功能、报警信息提醒功能、天气信息展示功能、资源信息展示功能、抢修值班人员信息展示功能、设备信息展示功能、实时数据信息展示功能、杆塔区域信息展示功能以及实时视频展示功能。

## 3.4 系统总体设计

系统由3个子系统构成，分别为边缘网关组态程序、业务服务程序、数据展示平台。

边缘网关组态程序系统部署在边缘网关中，通过Serial与AD采集终端进行通讯， 通过Socket与业务服务程序进行数据交互，业务服务程序部署在业务服务器中，通过Http协议与数据可视化终端进行数据交互，系统部署图如下图3-5所示。

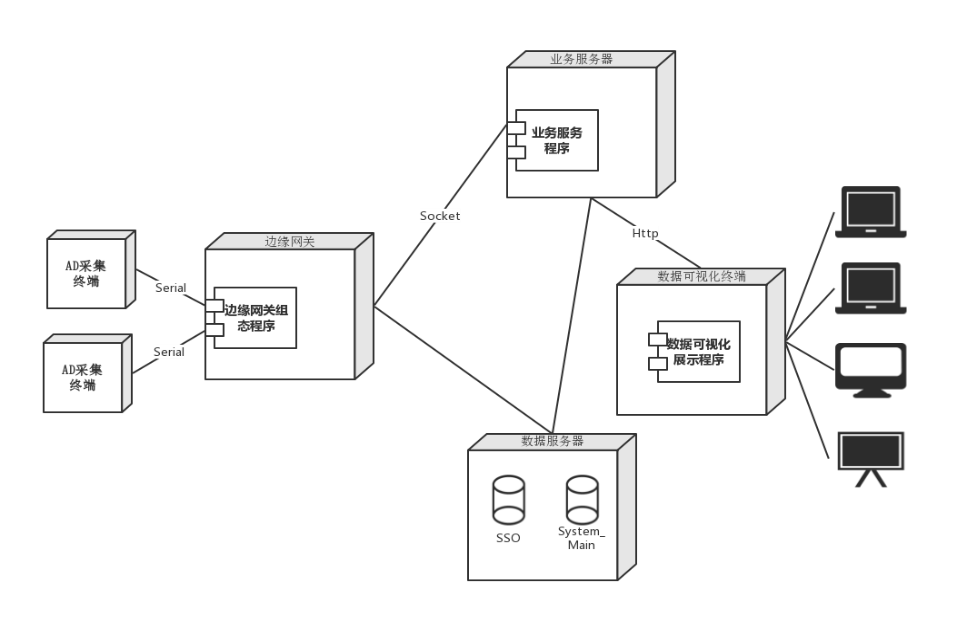


图3-5 系统部署图

本系统螺栓终端部署在高压杆塔上，通过ZigBee无线传输协议进行星型拓扑并传输至边缘网关，边缘网关将数据处理后分发至数据中心中的服务器中，数据中心服务器将根 可视化平台进行展示，报警信息直接展示在大屏中并触发报警，相应问题杆塔变色并传递信息至值班维修人员，导航至对应杆塔进行维修。

### 3.4.1 数据可视化展示平台设计

数据可视化展示平台基于Vue.js框架开发，使用Vue.js的双向数据绑定可以实现数据的异步展示，通过Echarts进行图表的绘制，通过腾讯地图提供的API进行地图展示，通过NURBS曲线进行变电站影响范围的绘制工作，数据可视化平台可以展示报警记录、实时报警信息、抢修资源等，通过使用对应视频的OCX控件结合环形缓冲队列的数据缓冲模式以实现视频的实时展示，通过WebSocket传输AD采集终端上传的实时数据。

### 3.4.2 业务服务程序设计

业务服务程序为基于Java Spring Boot框架编写的Web API，通过边缘网关连接的Socket与数据可视化展示平台连接的WebSocket由SocketManagement类进行管理，完成掉线设备、关闭设备的删除，新增设备的添加等操作，数据展示平台的设备命令通过WebSocket连接到服务程序后，通过EqumentManagement类解析后由Socket发送至对应边缘网关进行相应操作，同时执行相应的数据库操作。

报警类WarningMessag、实时信息类RealMessage、心跳信息类BeatMessage继承于Message类，当信号通过Socket接收后，由MessageManagement类中的DealMessage进行处理后通过JSONConvert转换后生成对应消息类型的对象进行下一步处理，类图如下图3-6所示。

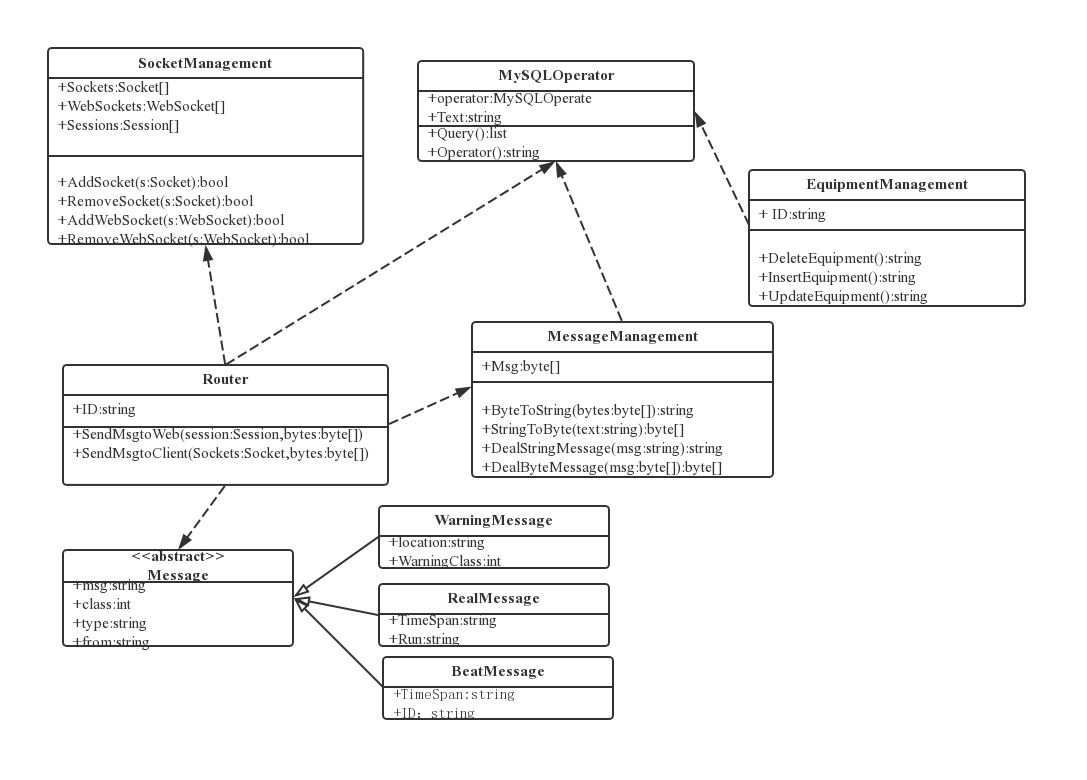


图3-6 业务服务程序类图

### 3.4.3 边缘网关组态程序设计

边缘网关组态程序为基于C#的WinForms程序，通过创建数据库连接池MySQLOperator类以异步操作数据的存储等操作，AD采集终端采集数据后，通过SerialRecive类将数据通过串口传递至边缘网关组态程序，程序通过cacheIfo进行数据缓冲后通过Router进行分发操作，轮询的数据通过MySQLOperator类中的OtherOperator()方法插入数据库中实时性数据通过Socket类中的Send()方法将数据发送至业务服务程序，系统类图如下图3-7所示。

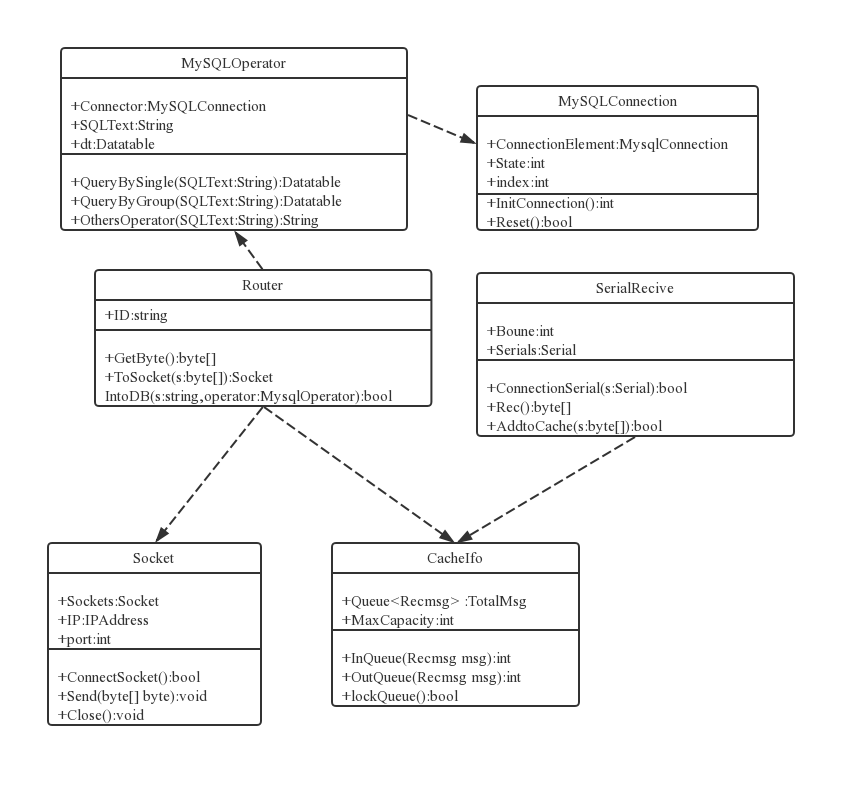


图3-7 边缘网关组态程序类图

## 3.5 数据库设计

本系统中以面向实体作为数据库对象的遴选原则创建16张数据表，分为2

个数据库。

### 3.5.1 System\_Mian数据库设计

System\_Main数据库服务边缘网关组态程序，业务服务程序，数据可视化展示平台，由10张表构成，分别为AD终端采集表、AD设备状态表、AD设备类型表、杆塔信息表、地区信息表、杆塔类型表、报警类型表、报警信息表、组态驱动表、监控设备信息表，System\_Main数据库E-R图如下图3-8所示。



图3-8 System\_Main数据库E-R图

AD采集终端表用于存储各个杆塔中AD采集终端的信息，在设备管理过程中，管理后台通过逻辑添加将设备信息预添加至本表中并将当前状态置为待部署，当设备通过边缘网关添加后，更新上线时间以及当前状态，添加之前系统会验证设备ID与待部署杆塔是否为同一杆塔，AD采集终端表如表3-1所示。

表3-1 AD采集终端表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Equipment\_id | char | 20 | 设备ID | 主键 |
| 2 | Equipment\_belongtower | char | 16 | 归属杆塔 | 外键 |
| 3 | Equipment\_type | char | 16 | 设备类型 | 非空 |
| 4 | Equipment\_date | date | - | 上线时间 | 非空 |
| 5 | Equipment\_state | int | - | 当前状态 | 外键 |
| 6 | Equipment\_location | varchar | 64 | 所在位置 | 非空 |

AD设备状态表用于映射AD设备的状态，AD采集终端表中的设备当前状态字段作为外键映射出本表，在设备状态变化过程中，设备状态码必须存在本表中，本表与AD采集终端表建立级联更新关系，AD设备状态表如表3-2所示。

表3-2 设备状态表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | State\_id | int | - | 状态ID | 主键 |
| 2 | State\_concent | varchar | 64 | 状态内容 | 非空 |

设备类型表用于映射AD设备的设备类型，本表与AD设备采集终端表建立级联更新关系。

表3-3 设备类型表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Equipment\_type\_id | char | 16 | 类型ID | 主键 |
| 2 | Equipment\_type\_name | varchar | 64 | 类型名称 | 非空 |

杆塔信息表用于存储杆塔信息，与AD设备信息表建立级联更新与级联删除，杆塔ID存储了部分信息，根据ID信息可以判断杆塔归属地等信息，杆塔信息表如下表3-4所示。

表3-4 杆塔信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Tower\_id | char | 20 | 杆塔ID | 主键 |
| 2 | Tower\_address | char | 19 | 杆塔地址 | 非空 |
| 3 | Tower\_connection | varchar | 64 | 杆塔连接 | 非空 |
| 4 | Tower\_type | char | 16 | 杆塔类型 | 外键 |
| 5 | Tower\_name | varchar | 64 | 杆塔名称 | 非空 |

地区信息表用于存储地区信息，杆塔信息表中的Tower\_id前6位为地区编码，根据地区编码可以获取分管辖分公司信息，本表作为权限分配中的映射表，地区信息表如下表3-5所示。

表3-5 地区信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Area\_number | char | 6 | 地区编码 | 主键 |
| 2 | Area\_name | varchar | 34 | 地区名称 | 非空 |
| 3 | Governing\_company | varchar | 34 | 管辖公司 | 非空 |

杆塔类型表用于存储杆塔类型信息，与杆塔信息表中的杆塔类型映射，根据杆塔类型的不同，其属性也不同，杆塔类型表如下表3-6所示。

表3-6 设备类型表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Tower\_type\_id | char | 16 | 类型ID | 主键 |
| 2 | Tower \_type\_name | varchar | 64 | 类型名称 | 非空 |

报警类型表用于存储报警信息类型，报警级别直接影响报警信息被边缘网关推送的顺序，报警级别最该高1，最低为3，边缘网关获取到最高等级的报警时将直接将报警信息推送至服务器，服务器直接推送到数据可视化平台，报警类型表如下表3-7所示。

表3-7 报警类型表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Warning\_id | char | 16 | 类型ID | 主键 |
| 2 | Warning\_name | varchar | 64 | 类型名称 | 非空 |
| 3 | Warning\_class | int | - | 报警级别 | 非空 |

报警信息表用于存储发生过的报警信息，报警信息由AD采集终端捕获由边缘网关生成，并传送至服务器与数据展示平台，报警信息表如下表3-8所示。

表3-8 报警信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Waring\_infoid | char | 20 | 报警ID | 主键 |
| 2 | Waring\_id | char | 16 | 设备ID | 非空 |
| 3 | Waring\_date | date | - | 报警时间 | 非空 |

续表3-8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 4 | Waring\_type | char | 16 | 报警类型 | 外键 |
| 5 | Waring\_concent | varchar | 64 | 报警内容 | 非空 |

组态驱动表用于存储各个AD采集终端的驱动程序，当组态程序通过设备OCX控件并发送请求至服务器时，业务服务程序将返回边缘网关组态程序设备驱动地址供其下载驱动，组态驱动表如表3-9所示。

表3-9 组态驱动表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Device\_id | char | 20 | 驱动ID | 主键 |
| 2 | Device\_address | varchar | 64 | 驱动地址 | 非空 |

监控设备信息表用于存储监控设备信息，监控设备新接入时将根据设备ID判断设备归属供应厂商，根据设备供应厂商下载对应视频设备驱动，并使用设备厂商提供的OCX控件显示设备提供的视频，根据设备的不同，对应的设备操作方式也不同，监控设备信息表如下表3-10所示。

表3-10 监控设备信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Video\_id | char | 20 | 设备ID | 主键 |
| 2 | Video\_type | char | 16 | 设备类型 | 非空 |
| 3 | Video\_tower | char | 16 | 设备杆塔 | 非空 |
| 4 | Video\_location | char | 16 | 设备位置 | 非空 |
| 5 | Video\_mac | varchar | 64 | 设备Mac | 非空 |
| 6 | Video\_ip | varchar | 64 | 设备IP | 非空 |

### 3.5.1 SSO数据库设计

SSO数据库服务于单点登陆业务，实现用户角色分配、系统的权限分配等功能，数据库由6张表构成，分别为用户表、系统信息表、权限信息表、角色信息表、角色权限表、用户角色表。SSO数据库E-R图如下图3-9所示。



图3-9 SSO数据库E-R图

用户表用于存储用户信息，用户ID为主键，用于唯一标识用户信息，用户密码存储为32位MD5密文，用户信息表如下表3-11所示。

表3-11 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | user\_id | char | 11 | 用户id | 主键 |
| 2 | user\_password | char | 32 | 用户密码 | 非空 |
| 3 | attribution\_company | varchar | 32 | 归属公司 | 非空 |
| 4 | ownership\_department | varchar | 32 | 归属部门 | 非空 |

系统信息表用于存储系统信息，单点登录时，系统编号作为用户是否具有该系统权限的唯一标志信息，系统信息表如下表3-12所示。

表3-12 系统信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | System\_id | char | 11 | 系统id | 主键 |
| 2 | system\_name | varchar | 34 | 系统名称 | 非空 |
| 3 | System\_specification | varchar | 34 | 系统描述 | 非空 |
| 4 | System\_icon | varchar | 34 | 系统图标 | 非空 |
| 5 | System\_link | varchar | 34 | 系统链接 | 非空 |

权限信息表用于存储权限信息，根据用户权限不同，其可以登录的系统与二级模块也不同，权限信息表如下表3-13所示。

表3-13 权限信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Permission\_id | char | 11 | 权限id | 主键 |
| 2 | Permission\_type | varchar | 34 | 二级权限 | 非空 |
| 3 | Permission\_description | varchar | 34 | 权限描述 | 非空 |

角色权限表用于存储角色与权限的映射关系，角色权限表如表3-14所示。

表3-14 角色权限表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Permission\_id | char | 11 | 权限id | 非空 |
| 2 | Role\_id | char | 11 | 角色id | 非空 |

用户角色表用于存储用户与角色的映射关系，用户角色表如表3-15所示。

表3-15 用户角色表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | User\_id | char | 11 | 用户id | 主键 |
| 2 | Role\_id | char | 11 | 角色id | 非空 |

角色信息表用于存储角色信息，用户登录系统时会根据角色映射到权限判断是否可以登录该系统，角色信息表如下表3-16所示。

表3-16 角色信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 数据库字段 | 字段类型 | 字段长度 | 存储内容 | 备注 |
| 1 | Role\_id | char | 11 | 角色id | 主键 |
| 2 | Role\_name | varchar | 32 | 角色name | 非空 |
| 3 | Role\_description | varchar | 32 | 角色描述 | 非空 |

第 4 章 关键算法设计

## 4.1 基于序列偏移化的设备合法性验证算法

由于系统下位AD采集终端与上位边缘网关采用无线方式连接，若下位接入设备不符合要求或下位AD采集终端被不法分子伪造，将会造成杆塔大面积故障，因此本系统采用了设备合法性验证加密算法验证设备是否符合接入规范与合法性。

边缘网关将始终暴露一个不参与任何数据传输的串口供新设备连接使用，新下位AD采集终端启动时，将自动访问该公共端口，为防止由于杆塔距离过近导致设备连接其他杆塔的边缘网关，设备接入前需要在管理后台进行预先添加，当新设备接入，边缘网关将会根据设备ID判断该设备是否属于该杆塔需要连接的设备，如不是，系统将会拒绝该设备的连接请求，若该设备合法，将进入序列偏移化验证阶段。在设备偏移化验证阶段，首先将由边缘网关生成36位随机字符序列与4位随机数并发送至待连接的AD数据采集终端，终端接收到信息后将通过序列偏移化算法进行加密，加密后将密文传输至上位边缘网关进行验证，边缘网关在验证时同样会将原序列进行偏移并比对偏移后的序列是否相同，若比对相同，网关将设备信息存储至数据库与内存中并开始接收该设备的数据并对其进行控制，合法性验证过程流程图如下图4-1所示。



图4-1 设备合法性验证流程图

为保证设备连接不被伪造，系统将定时重新生成随机参数对下位设备进行二次验证以保证连接在边缘网关的设备始终为合法设备。

### 4.1.1 随机数生成算法

本算法中，需要随时生成4位随机数，算法会根据随机数的不同对原序列进行不同方式的偏移，因此随机数的生成必须要均匀，本随机数生成算法采用线性同余法生成随机数可以保证生成的随机数在[0,1]内是均匀随机的。根据式(4-1)所述的线性同余法可以生成随机数。

(4-1)

其中为一个随机数列，其中为随机种子，为一个非负整数，和为两个互质的正数，为与互质的质数。随机数生成方法如下所示。

public static long GenerateRandomNumber()

{

long ExcessVariable = 2147600009 \* Seed + 2147599931;

Seed = ExcessVariable % 2147483629;

return Seed;

}

在本方法中，生成4位随机数需要将该方法迭代4次即可，为验证上述算法生成的随机序列的随机性是否符合均匀随机值，本系统利采用了OpenGL对上述算法生成的一万次随机生成序列与随机种子的关系进行了绘制，该散点图用于验证随机性，利用OpenGL生成的散点图如下图4-2所示。该散点图横坐标为随机种子值，纵坐标为生成值，依据散点图，该算法生成的随机数较为均匀。

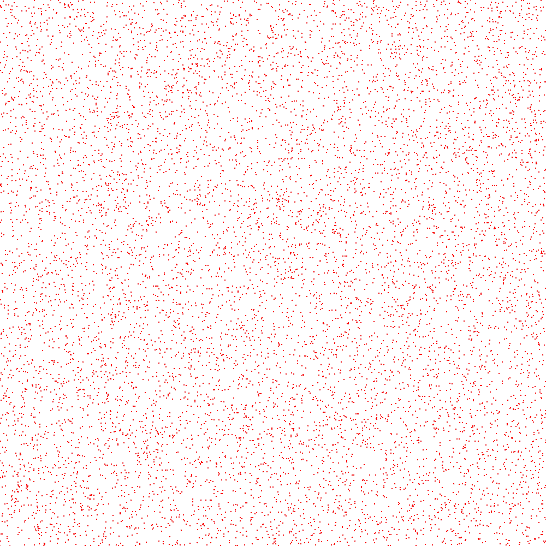


图4-2 随机数散点图

### 4.1.2 序列偏移化算法

本系统进行设备验证的过程中，核心算法为序列偏移化算法，系统生成36位随机序列后进行序列偏移化时首先通过ASCII码变换将所有字符序列转换为36位纯数字。

原字符序列转换为数字后，从首个取出数字起，每次取1个数字，并以斐波那契数列为步长取8个数字并求和，使用斐波那契数列作为步长可以使得距离当前取得位较近的数据产生较大影响，防止各个数据因偏移权重相似而造成偏移结果相同或相近。

随机生成的四位随机参数进行组合计算，经过首次偏移后的序列随机选择组合后的新参数进行求和，生成的新数字依次进行模62取余计算以保证新生成的字符序列为字母与数字组合，新生成的偏移序列已具有相当的混乱程度且通过取余运算以不可逆转，可以进行设备合法性验证，在设备合法性验证过程中，实际为验证设备内存储的算法是否一致，进行验证时，若随机数改变，也会造成生成的偏移序列具有较大差别，算法流程图如下图4-3所示。



图4-3 序列偏移化算法流程图

## 4.2 实时信息流推送权限管理

本系统实时数据交互通过TCP全双工通讯实现，边缘网关组态控制程序与业务服务器采用Socket连接，数据展示平台与业务服务器间采用WebSocket长连接。

当新边缘网关启动时，边缘网关将连接服务器暴露的统一连接端口进行连接，服务器将会根据端口连接情况将新套接字信息发送到边缘网关，边缘网关接收信息后与分发端口断开连接并尝试与新端口连接，新端口接收到连接请求将对待连接的设备进行身份认证，经过序列化偏移验证成功后的设备，将保持与服务器的长连接，服务器将对应边缘网关ID与连接的Socket对象存储之Redis中，流程图如图4-4所示。



图4-4 边缘网关Socket连接流程图

当用户登录数据展示平台时，数据展示平台将与服务器建立WebSocket连接，此时不需要进行连接验证，系统会根据权限选择展示的杆塔或变电站信息，用户信息将存储在Session中并由服务器保存在Redis中。

当报警信息或实时数据由AD采集终端产生后，首先通过Socket上传至业务服务器，业务程序通过Socket对象判断上传数据的网关对应权限并将信息推送至具有该网关查询权限的数据可视化展示平台。

## 4.3 杆塔连线渲染算法

本系统数据展示平台利用腾讯地图API进行地图的渲染、杆塔及杆塔间的连线，由于杆塔拓扑的连线是网状的，进行杆塔连线渲染时，杆塔之间连线易出现重复绘制的情况，因此本系统通过连线渲染算法对系统连线节点进行优化，使得连接渲染时不再重复绘制。

数据可视化平台初始化加载时或地点选择改变时，数据可视化展示平台将向业务服务发送查询请求，查询指定位置周围1000米内的杆塔数据全部返回，服务器以Json的格式返回的杆塔信息包括一个名为“tower\_connection”的字段，该字段通过“-”区分该杆塔的前驱杆塔与后置杆塔，如

tower1，tower2 - tower3

即表示该杆塔的前驱杆塔分别为“tower1”与“tower2”，后置杆塔为“tower3”。

系统获取对应杆塔连接信息后开始进行点对点绘制，使用腾讯地图API进行电的绘制，代码如下所示。

var cirle = new qq.maps.Circle({

center: new qq.maps.LatLng(39.920, 116.405),

radius: 2,

map: map

});

在绘制开始前系统会创建一个空的数据字典用于存储已绘制的点，每绘制一个杆塔，系统都会将已经绘制的杆塔序列对存入字典，而后每当新取出杆塔，系统都将会判断新生成的序列对是否存在与字典序列中以减少重复绘制的次数。判断连接序列对是否存在的代码如下所示。

function havetheLine (a, b)

{

if (a.length !== b.length) return false

let c = b.slice()

for (let i = 0, len = a.length; i < len; i++) {

let j = c.indexOf(a[i])

if ( j === -1) return false

c.splice(j, 1)

} // 删除已经匹配的元素，可以缩短下次匹配的时间

该算法运行流程图如图4-5所示。



图4-5 杆塔连线渲染算法流程图

## 4.4 高并发信号缓冲机制

由于本系统终端数量较多，产生的数据也较为庞大，当发生突发紧急状况时，可能存在大量数据同时发送到边缘网关的情况，边缘网关的数据处理速度无法平衡接收数据的速度则会导致待处理数据滞留，因此，本系统采用路由分发，分类处理的方式，系统数据流图如图4-6所示。



图4-6 缓冲算法数据流图

本系统规定了固定的数据报文格式并打包成Json格式进行传输，从下位机上传的数据只需要按照规约进行转换边缘网关即可，报文格式如图4-7所示。



图4-7 报文格式

数据类型标识中包含上传数据的AD采集终端类型标识，数据类型等信息，当边缘网关接收到数据后，首先判断网关是否已经下载了对应的解释驱动，如网关内没有数据驱动，则会根据数据类型标记中的终端类型标识进行查询，将对应数据解释驱动下载到网关中并解释数据。数据域中包含了类对象转换为JSON的字符序列，并使用AES加密算法进行加密，只有符合规定的设备才能解释数据。

数据类型标识不采用AES加密便于边缘网关直接判断数据类型并路由至相关解释驱动，边缘网关接收到数据后仅对数据类型标识进行解析，加密的类对象直接转发。

边缘网关启动时，首先创建相应的缓冲区，代码如下所示。

public static int MaxCapacity = 1500; //缓冲结构最大容载量

public static Queue<RecMsg> TotalMsg = new Queue<RecMsg>();//总缓冲区

数据通过端口进入边缘网关对应端口将直接放置在缓冲区中，入队时队列加锁，放入完成后解锁，等待取出线程对队列进行操作，当队列满队时，系统自动删除队头优先级较低的数据，代码如下所示。

public static void EnterArea(RecMsg rec)

{

lock (TotalMsg)//队列加锁

{

if (TotalMsg.Count >= MaxCapacity)

{

TotalMsg.Dequeue();

}

TotalMsg.Enqueue(rec);

}

}

当数据进入缓冲区后，提取线程开始工作，将队列加锁并从缓冲区队列中取出数据交由下一级缓冲处理，代码如下所示。

public static void OutArea()

{

lock(TotalMsg)

{

if(TotalMsg.Count != 0)

{

Router.AreatoFactoryRouter(TotalMsg.Dequeue());

//交给一级路由

}

}

}

数据提交到下一级别缓冲区域前需要先通过路由判定类别进行分发，系统将下位传递的信息根据优先级分类，分类后存进相应缓冲区，代码如下所示。

public static void AreatoFactoryRouter(object msg)

{

switch (((RecMsg)msg).Priority)

{

case 1:

Factory.EnterFactory(

Factory.MsgList\_First, (RecMsg)msg);

break;

case 2:

Factory.EnterFactory(

Factory.MsgList\_Second, (RecMsg)msg);

break;

case 3:

Factory.EnterFactory(

Factory.MsgList\_Third, (RecMsg)msg);

break;

}

}

进行二级分类后的数据将由三个线程分别取出交由下一级缓冲，此时将同优先级的数据进行信息分类，通过报文头中未加密的部分进行分类，代码如下。

public static void FactorytoDataOperator\_First(object msg)

{

if (Factory.MsgList\_First.Count > 0)

{

switch (((RecMsg)msg).SignalType)

{

case "A":

DataOperator.OperateFirstQueue((RecMsg)msg);

break;

case "B":

DataOperator.OperateSecondQueue((RecMsg)msg);

break;

case "C":

DataOperator.OperateThirdQueue((RecMsg)msg);

break;

}

}

}

数据经过再次分类后最终将交由对应算法进行处理，上述算法分别通过优先级以及数据类型进行分类，可以保证优先级较高的数据优先被处理，如报警信息等；较低优先级并且属于C类可抛弃数据可以在缓冲区爆满时进行必要舍弃，如过期的正常数据或心跳包，该方法可以解决数据并发量较大时通过数据治理减少信息处理不及时的问题。

## 4.5 基于非均匀有理B样条的闭合区域绘制

本系统在对变电站归属区域的杆塔进行范围绘制时，由于腾讯地图API不提供曲线绘制且采用普通曲线无法完成矢量图的绘制，本系统采用非均匀有理B样条来绘制区域包围线，相比传统贝塞尔曲线而言，非均匀有理B样条可以更好的控制由多个控制点控制的曲线，可以控制曲线的部分而不需要考虑各个控制点对曲线整体的影响，曲线如图4-8所示。

图片包含 天空, 户外

描述已自动生成

图4-8 NURBS曲线

数据可视化展示平台加载或区域切换时，系统通过坐标拾取器选择杆塔周围的5个坐标点作为曲线的控制点，通过计算节点集对其中的点进行绘制。

计算节点集前需要计算每两个控制点间的距离，在本方法中根据两点之间距离公式依次计算相邻两点的距离，为减少不必要的浮点运算，本方法将每一个距离值转换成整型。在计算完每相邻两个控制点的距离后，根据哈德利-贾德法计算节点集U。

哈德利-贾德法只计算部分节点值，其余节点根据哈德利-贾德法推导公式已知,节点值将直接影响曲线连续性和光滑性，为保证数值的精确性，本方法使用双精度浮点型储存节点值。

在完成节点值的计算后，根据节点值推导K阶B-样条曲线基函数，三阶基函数生成的曲线已经能够满足需求。因需要使用三阶基函数计算曲线上点，在本方法中通过将该方法递归三次来获得三阶基函数。为防止计算数据出现异常，当节点值差距过小时，基函数视作0。

在完成基函数和节点的计算后，可将计算出的基函数和控制点坐标带入曲线函数计算出曲线上的任意点，根据曲线公式逐个计算出曲线上的点，通过控制参数u的步长决定共需要计算出点的坐标，即坐标光栅化，最后通过腾讯地图API点的绘制对曲线进行绘制，算法流程图如下图4-9所示。



图4-9 NURBS算法流程图

系统通过腾讯地图API渲染杆塔影响范围区域的核心代码如下

var path = []

var point

for(var i in a){

point = new qq.maps.LatLng(a[i][0],a[i][1])

path.push(point)

}

var polygon = new qq.maps.Polygon({

clickable: true,

cursor: 'crosshair',

editable: false,

fillColor: new qq.maps.Color(100, 50, 150, 0.5),

map: map,

path: path,

strokeColor: '#000000',

strokeDashStyle: 'solid',

strokeWeight: 5,

visible: true,

zIndex: 1000

});

第 5 章 系统功能的设计与实现

## 5.1 边缘网关组态程序

本系统中每个杆塔下方均配备一台具有Windows操作系统的PC机作为边缘网关，边缘网关组态程序为基于.Net Framework 4.7框架的WinForms程序，可以完成数据中转、处理，报警推送、自动巡检等功能，程序主页面如下图5-1所示，具有通信参数配置、终端信息显示、终端操作、服务配置等功能。

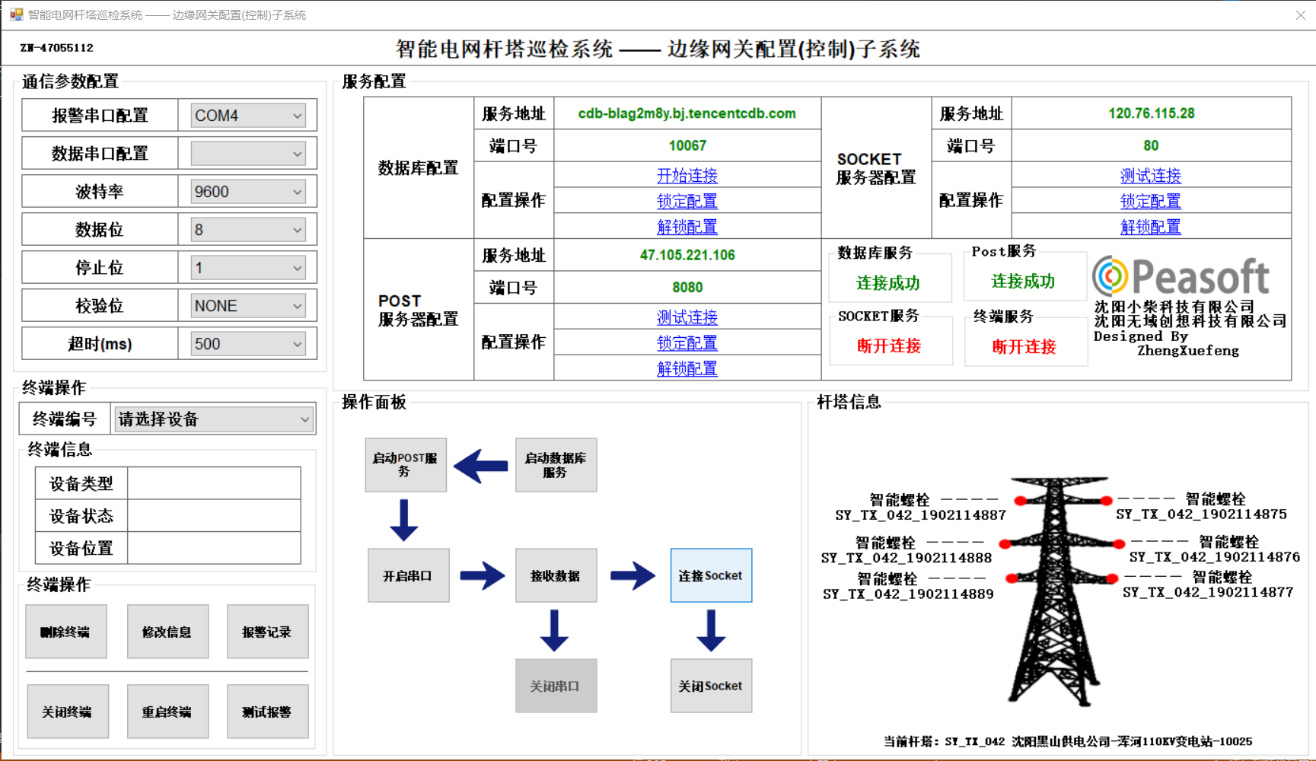


图5-1 边缘网关组态程序

### 5.1.1 通信参数配置模块

本系统下位AD采集终端与上位边缘网关采用无线串口通讯，报警与数据传输分别由两个不同串口负责以增进报警信息的传递效率，系统配置串口信息时提供串口缺省值，串口配置函数由无参配置方法和有参配置方法两个重载函数构成，无参配置方法代码如下

public void OpenPort()

{

try

{

if (PortNameArr.Length > 0 && !ports.IsOpen)

{

ports.PortName = PortNameArr[0];

ports.BaudRate = 9600;

ports.DataBits = 8;

ports.StopBits = (StopBits)1;

ports.ReadTimeout = 500;

ports.Open();

}

}

catch(Exception exp)

{

throw exp;

}

}

该方法中默认波特率为9600，数据位为8，停止位为1，超时阈值为500毫秒，在接入设备没有特定需求时，系统均以默认配置运行。有参配置方法如下

public void OpenPort(string portName, int boudRate, int dataBit, int stopBit, int timeout)

{

if (!ports.IsOpen)

{

try

{

ports.PortName = portName;

ports.BaudRate = boudRate;

ports.DataBits = dataBit;

ports.StopBits = (StopBits)stopBit;

ports.ReadTimeout = timeout;

ports.Open();

}

catch (Exception exp)

{

throw exp

}

}

}

该方法中，串口名称、波特率、数据位、停止位、超时阈值均需要自行设定，用于特定的设备配置。

### 5.1.2 终端操作排查模块

由于系统板载电池或终端存在使用寿命，因此在板载电池供电不足时或终端故障时需要人工进行更换，工程人员可使用终端操作模块查询故障或需要更换的设备信息，异常设备会使用红色进行提示，如下图5-2所示。



图5-2 设备信息显示

终端操作中提供了测试功能，如测试报警等，测试报警采用专用报文格式，优先级仅低于正式报警，用于测试使用，测试报警信息由服务器截断，不再向数据展示平台发送，截断后向组态程序发送成功信息，完成报警的测试流程以防止数据展示平台因测试报警展示而降低报警接收效率。

### 5.1.3 信息配置模块

由于不同地域的杆塔所连接的数据中心也不相同，因此边缘网关组态程序提供信息配置功能，分别连接不同的数据库、业务服务器以及Socket服务器，除数据库外，其他服务器在接入边缘网关组态程序时，将判断该网关的唯一标识码，确认合法连接后开始接收数据或转发数据，若连接指向错误，服务器发送正确的连接指向到组态程序，帮助其连接正确的服务器。

### 5.1.4 系统组态

由于本系统的边缘网关需要处理部分数据，因此边缘网关需要加载AD采集终端的驱动程序，接入本系统的AD采集终端驱动不尽相同，其使用驱动以及解释程序也有所不同，当新类型设备接入系统，系统将加载对应的组态OCX控件以获得对应驱动以及位置信息。

系统获得对应位置传输的信息时，会根据对应的控件驱动进行路由分发，交由对应驱动进行信息处理，组态配置程序如图5-3所示。

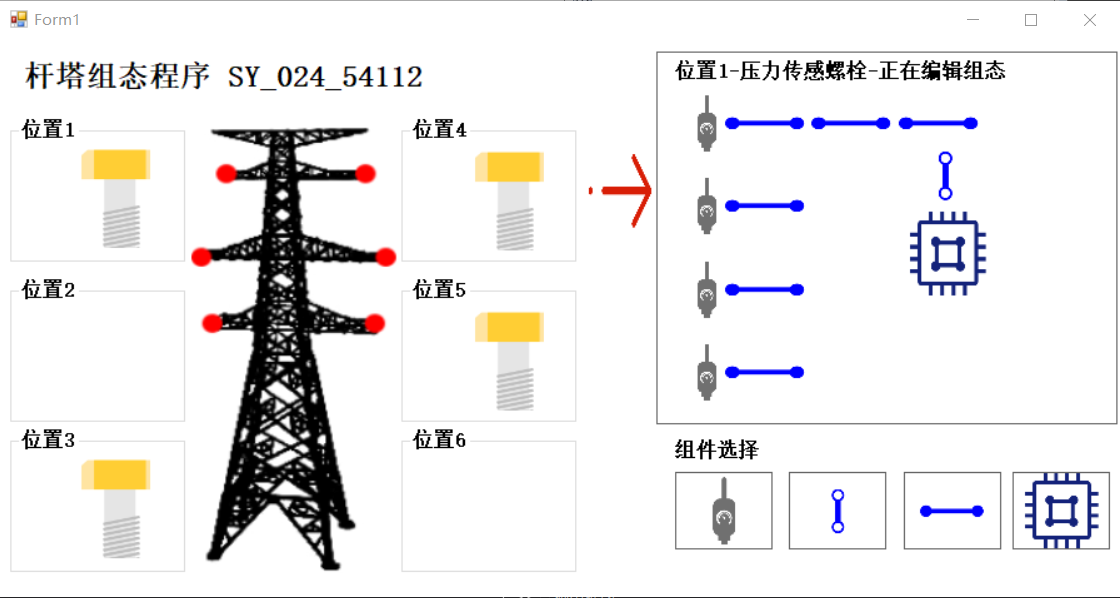


图5-3 组态配置程序

## 5.2 数据展示平台

本系统数据展示平台具有实时报警提示、历史报警记录查询；天气信息预警；抢修资源，人才资源显示；地图杆塔显示；杆塔信息、设备信息可视化展示；设备信息实时展示；杆塔监控视频展示等功能，系统主页面如下5-4所示。



图5-4 数据展示平台主页面

### 5.2.1 单点登录

本系统存在多个子系统以及对应多个权限，本系统采用SSO单点登录对系统进行登录，登录权限判定由权限与角色构成，权限仅对于系统而言，一个权限对应一个系统，一个角色对应多个系统权限，因电网部门为聚集型权限分配模式，即一个部门同级所有人员的权限大致相同，且人数众多，采用角色对应更方便管理。单点登录主页面如图5-5所示。



图5-5 单点登录页面

用户登录时，首先用户进入SSO单点登录页面,输入手机号点击获取验证码 此接口不予拦截，并在此间判断了用户是否注册，如未注册不予发送验证码。如用户已经注册过 此时将验证码发送至请求用户的手机中，验证码3分钟有效。

一个用户可有多种角色，每一个角色可有多种权限，新注册用户默认为无角色即为无权限 需要向管理员申请分配，角色可动态可定制被分配权限，用户可动态定制角色，用户被系统判定登录成功后根据用户id查找该其角色并根据角色id查询权限即可得出该用户的所有权限，当新用户使用该账号登录时，新的Session将会取代过期Session，使原登录用户下线，并给予相应短信提示，如图5-6所示。

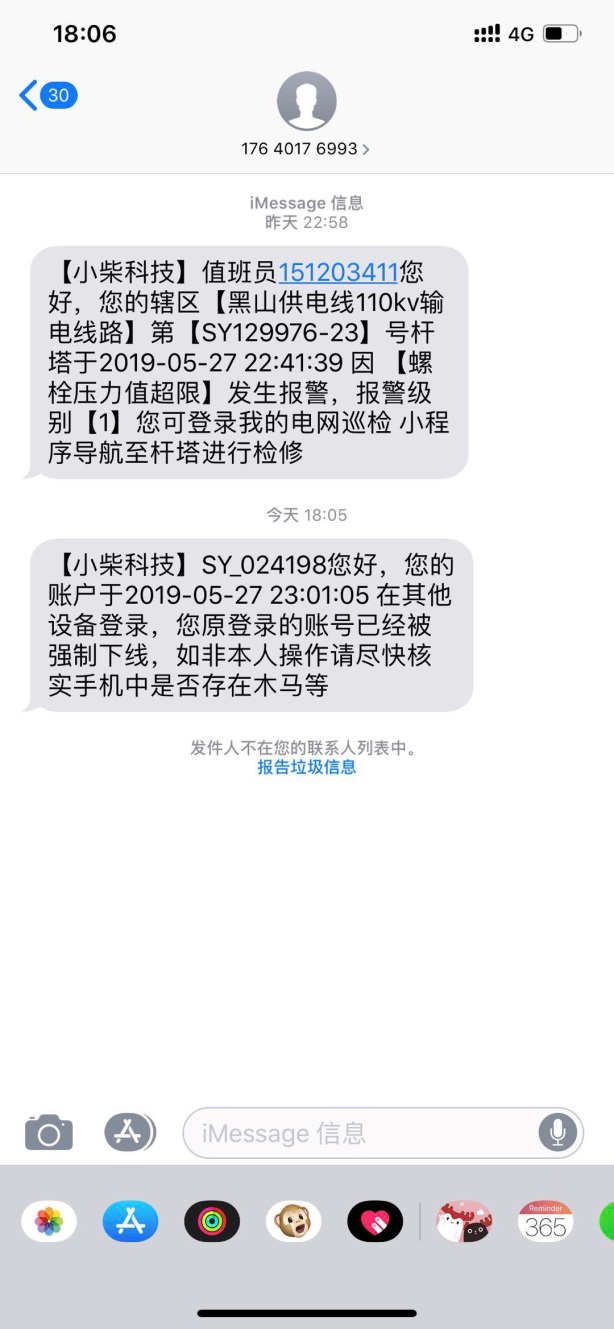


图5-6 异地登录提醒

用户登录成功后登录系统将根据权限判断用户可登录哪些系统并显示在菜单中，用户选择对应系统即可进入对应系统，如图5-7所示。



图5-7 系统选择页面

### 5.2.2 报警提示模块

由于本系统实时报警提示模块与服务器采用WebSocket连接，每当数据展示平台接收到报警信息时，系统自动向对应值班人员发送报警提示短信，如图5-8所示。

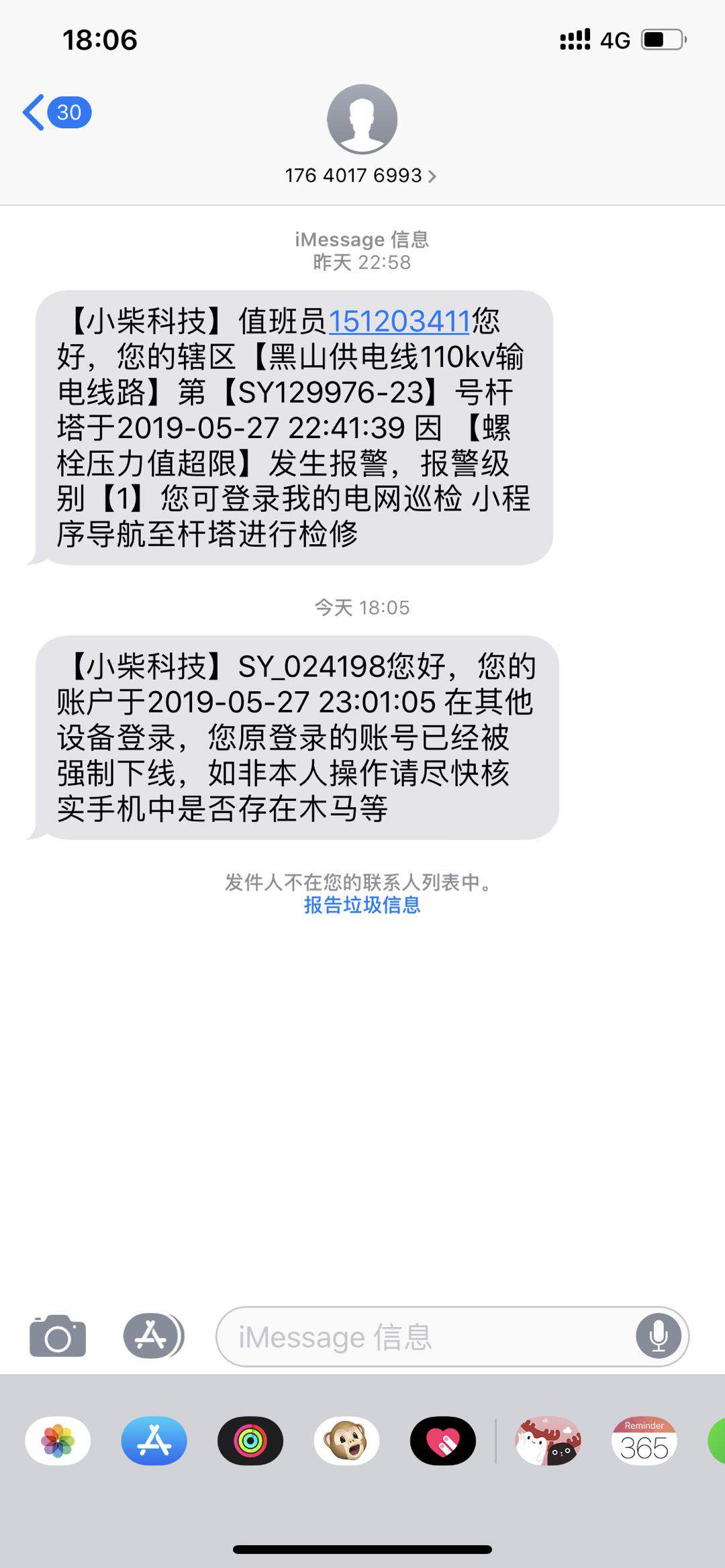


图5-8 报警提示短信

短信推送部分代码如下所示

public static void Sender(List<string> templParams, string phone , ref string result)

{

QcloudSms.SmsSingleSenderResult singleResult;

QcloudSms.SmsSingleSender singleSender = new QcloudSms.SmsSingleSender(Config.Getappid(), Config.Getappkey());

singleResult = singleSender.SendWithParam(phone, Config.Gettmplateid(), templParams);

result = singleResult.ToString();

}

系统通过短信内置的模板进行参数添加并提交到对应接口进行短信发送。同时，数据展示平台将弹窗报警，并伴有报警提示警报音，如图5-9所示。



图5-9 报警提示窗

数据展示平台左上侧同时将展示最近3条报警记录，点击对应报警记录即可查看目前维修人员派遣状态、维修状态等信息，如图5-10所示。



图5-10 近三次报警记录

点击报警记录，即可查看所有报警记录，点击对应报警记录，即可查看每次报警的处理情况，人员派遣信息等，如图5-11所示。



图5-11 报警记录

### 5.2.3 抢修人才、资源展示模块

每次更改城市选项，数据展示平台均会向服务器发送请求，询问当前城市拥有的抢修资源、抢修人才种类和数量。数据展示平台接收到回传信息对数据进进行分类处理通过Echarts组件库中的甜甜圈图展示。抢修资源、人才资源展示模块如图5-12所示。

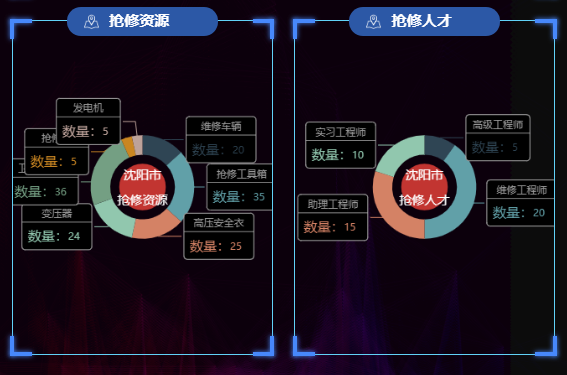


图 5-12 抢修资源、人才展示模块

Echarts绘制代码如下

var myChart = echarts.init(

document.getElementById('chart\_b'));

$.ajax({

url:'http://120.76.115.28:8080/TowerEquipmentBasicInform/count',

type:"GET",

success:function(result){

var option = {

legend: {

data:['高级工程师','维修工程师','助理工程师','实习工程师'],

},

series : [{

type:'pie',

data:[

{value:result.reluso[3].count, name:'高级工程师'},

{value:result.reluso[1].count, name:'维修工程师'},

{value:result.reluso[2].count, name:'助理工程师'},

{value:result.reluso[1].count, name:'实习工程师'}

]

}]

};

myChart.setOption(option);

})

### 5.2.4 杆塔区域展示模块

杆塔区域展示模块通过腾讯地图API的覆盖物进行杆塔点的绘制并通过杆塔渲染方法进行连线渲染，地图上方有地理位置关系三级联动选择器，可以选择查询城市。选择城市后地图中心点会移动到所选城市中心。

地图通过中心点坐标渲染地图代码如下所示。

map = new qq.maps.Map(document.getElementById("chart\_map"), {

center: new qq.maps.LatLng(41.792993,123.395605),

zoom: 30,

mapTypeId: qq.maps.MapTypeId.HYBRID,

panControl: false,

zoomControl: false,

noClear: false,

scrollwheel:false//滚轮放大

})

通过选择对应变电站可以查看变电站所管辖区域内杆塔的信息。如图5-13所示。

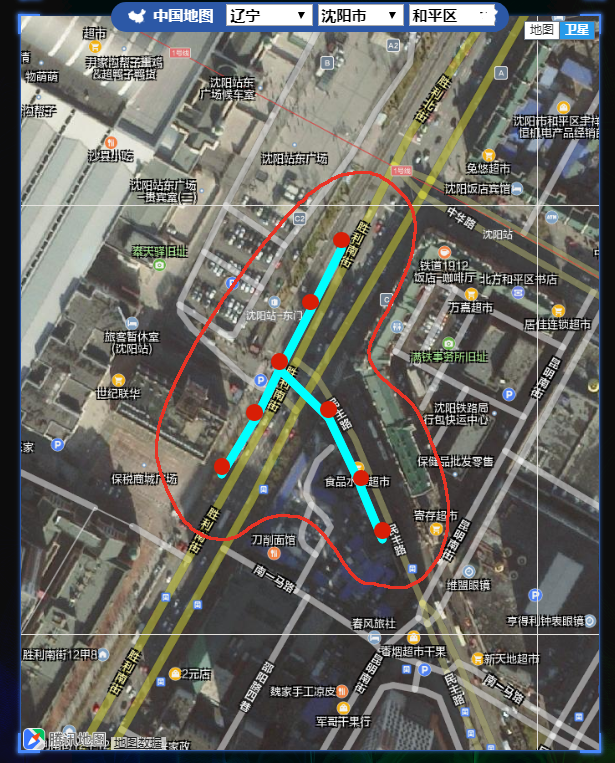


图5-13 杆塔区域信息展示

系统详情可以获取该杆塔详细信息，也可以获得对应设备的实时数据，实时数据通过EChart绘制出实时曲线，如图5-14所示。



图5-14 终端实时数据展示

系统可根据对应杆塔展示对应杆塔中的监控信息，监控采用海康威视网络摄像头实现，通过加载对应OCX控件将摄像头实时视频展示在数据展示平台中，摄像头OCX控件加载代码如下所示。

//调用OCX接口方法要传递的参数变量

var localIp = '172.16.100.9';//本地IP配置

var httpIp = '172.16.100.65';//通信IP

var httpPort = 7506;//通信端口

var userName = 'role\_ywy';//通信用户名

var pswd = '36e2c010de306ba7fbfff53b02bd082b';//通信密码 输入明文的MD5值

var scamera = '098810000003020001';//视频播放的设备编号

var userCode = '110010000004000155';

var videoMainWndIndex;

var bExit = true;

var parent;

var strDraw;

ocxHtmlLoad();

监控视频流在数据展示平台中效果如下图5-15所示。



图5-15 杆塔视频监控

系统报警后，本系统会将报警信息、杆塔信息等数据直接发送至对应的值班人员使用的“我的电网巡检”小程序中，电力维修人员可以根据小程序的导航指引到达对应故障杆塔维修设备，小程序如图5-16所示。



图5-16 我的电网巡检小程序

第 6 章 结 论

本研究基于“泛在电力物联网”的思想，设计并开发了以解决杆塔巡检难为目的的高压杆塔螺栓自动巡检系统。系统由边缘网关组态程序、业务逻辑服务与数据可视化展示平台构成，边缘网关组态程序提供了螺栓终端配置、杆塔信息管理、实时数据采集、实时数据分析、报警信息实时推送等功能；业务服务程序提供了Socket管理、各类信息管理、报警信息转发等功能；数据可视化展示平台提供终端实时数据展示、报警提醒、天气信息展示、值班人员管理、终端管理、实时视频监控等功能。经试用单位测试，结果表明系统运行稳定，功能满足应用需求，可实现杆塔自动巡检的要求，达到了预期设计目标。

系统数据展示平台采用的Vue框架可以有效的避免数据刷新过快，DOM操作较多导致的刷新延迟及加载速度过缓的问题。利用“非均匀有理B样条”绘制的区域边界曲线有效的解决了贝塞尔曲线因控制点过多导致的边界绘制不准确的情况。采用三级数据缓存路由方案可以防止由于数据量过大导致的报警信息传递不及时的情况，以3000条实时数据进行测试，报警信息全部被优先捕获并处理。采用了基于偏序覆盖的方式解决了地图API对杆塔连线进行绘制时因重复绘制连接线而导致的渲染缓慢问题，使得地图渲染性能提高了近50%。采用Spring Boot框架编写的API服务，使得数据展示平台与服务器之间采用接口服务的形式交换数据，降低了系统结构的耦合度，并避免了将应用层直接暴露给用户所导致的数据安全问题。

系统试用效果良好，运行稳定。但目前AD数据采集终端与边缘网关链接时对接入设备合法性的鉴别方法仍未能有效防止“中间人嗅探”等非法攻击手段，可通过增加对接入设备的二次签名校验解决该问题，这将是系统进一步完善的工作重点。

。

在学研究成果

[1] 2018年全国大学生计算机设计大赛辽宁省二等奖

[2] 2018年腾讯杯微信小程序设计大赛东北赛区三等奖

[3] 2018年大学生创新项目省级立项

[4]企业试用报告2份

参考文献

1. 陈麒宇. 泛在电力物联网实施策略研究[J/OL]. 发电技术: 1-8[2019-05-28].http://kns.cnki.net/kcms/detail/33.1405.th.20190524.1111.004.html.
2. 王哲, 孙博宇, 李传号. 采空区输电线路杆塔倾斜在线监测系统[J]. 电子技术与软件工程, 2014, (05): 159-159.
3. 翟雪奎, 张轶. 智能电网输电安全巡检系统的设计与开发[J]. 地理空间信息, 2014, 21(01): 44-45.
4. 王经. 能源供给侧结构性改革与能源互联网[J]. 上海节能, 2016, (12): 641-646.
5. 杨志武. 基于压电阻抗技术的螺栓松动监测的研究[D]. 武汉: 武汉科技大学, 2015.
6. 姚敏强, 杨坤怡. 基于图像识别的智能螺栓拧紧机的研究[J]. 机械设计与制造, 2008, 21(05): 127-129.
7. 田涛, 黄冰, 黄祖荣, 等. 基于MSP430F149的无线遥控智能螺栓紧固器[J]. 电气自动化, 2016, 38(04): 42-44.
8. 周文魁, 李晶. 江苏智能电网产业发展研究[J]. 中外企业家, 2015, 38(34): 68-69.
9. Andrzej Nowakowski, Paweł Wlazło, Radosław Przybysz. Intelligent Current Sensors as Part of Smart Grid Network[J]. Acta Energetica, 2015, 23(2).
10. 沈华韵, 张鹏, 王侃. 改进线性同余法随机数发生器[J]. 清华大学学报(自然科学版), 2009, 49(02): 191-193.
11. Felicisimo V Wenceslao Jr. Enhancing the Performance of the Advanced Encryption Standard (AES) Algorithm Using Multiple Substitution Boxes[J]. International Journal of Communication Networks and Information Security, 2018, 10(3): 35-38.
12. 魏琳, 田波. 基于STM32F4系列的串口DMA数据处理传输研究[J]. 自动化应用, 2016(08): 92-93.
13. 孙明刚. 一种使用环形缓存和环形队列实现UDP高效并发的方法[J]. 中国新技术新产品, 2016(11): 6-7.
14. Paul Krill, Paul Krill. Vue.js lead: Our JavaScript framework is faster than React[J]. InfoWorld.com,2016, 21(03): 181-183.
15. Varshney Nitish, Aggarwal Sumit, Kumar Shalabh, Singh S P. Retention and patient satisfaction with bar-clip, ball and socket and kerator attachments in mandibular implant overdenture treatment: An in vivo study[J]. Journal of Indian Prosthodontic Society, 2019, 19(1): 5

致 谢

大学四年的充实生活即将结束，在此论文即将完成之际，在这里我要向对我这篇论文提供了巨大帮助的老师、公司、同学们表示感谢。

首先我要感谢我的导师邵中老师，这篇论文的完成，离不开邵老师的悉心指导。同时也要感谢软件学院全体老师的教学，使我系统掌握了软件及软件工程的专业知识，为完成此论文打下了扎实的基础。

感谢理学院的凌楠、张刘阳、管林同学为我的这篇论文以及系统提供了算法方面的支持。

感谢国网辽宁省电力有限公司提供了数据支持，以及辽宁省电力有限公司信息通信分公司提供的技术协助。