

用电设备运行状态监控系统设计与实现

周 晓, 田瑞清, 李永清

(浙江工业大学 信息工程学院, 杭州 310023)

摘要: 用电设备工作电流在一定程度上能反映设备运行状况, 所设计的基于电流检测的用电设备运行状态监控系统中, 采集节点采用 STM32F411 作为主控芯片, 以新型数字型电流检测芯片 HLW8032 采集用电设备电流, WiFi 收发模块连接路由接入网络后, 将电流数据传输到云服务器的数据库保存, 云服务器采用轻量级企业集成框架 Struts+Hibernate+Spring+DWR+JQuery 开发, 使用 JAVA 的面向对象设计模式; 用户通过浏览器登录系统可查看用电设备的运行状态、使用时长、过载报警情况等, 实现对用电设备远程实时监控; 实验证明: 系统运行稳定, 能实现对用电设备电流值的准确采集。

关键词: 用电设备; STM32F411; HLW8032; WiFi; 监控系统。

Design and Implementation of Running State Monitoring System for Electrical Equipment

Zhou Xiao, Tian Ruiqing, Li Yongqing

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: The working current of the electric equipment could reflect the operation condition of the equipment to a certain extent. In the design of electric equipment running state monitoring system based on current detection, STM32F411 was used as the main control chip in the acquisition node. By using a new type of digital current detection chip HLW8032 to collect the current of electric equipment and connecting the WiFi transceiver module to the route and access to the network, storage of current data was transferred to the database of the cloud server. The cloud server was developed with the lightweight enterprise integration framework Struts+Hibernate+Spring+DWR+jQuery, and the object-oriented design pattern of Java was used. Through the browser login system, users can check the running status of the power equipment, use the time, overload alarm and so on, to realize the remote real-time monitoring of the power equipment. The experiment proves that the system runs stably and can reliably collect the current value of the electrical equipment.

Keywords: electrical equipment; STM32F411; HLW8032; WiFi; monitoring system

0 引言

随着我国经济的迅速发展, 用电设备得到广泛使用, 用电设备高效管理成为了重要的课题。由于计算机网络技术的高速发展, 尤其是 Internet/Intranet 技术的广泛应用, 为基于网络的设备管理系统的实施提供了坚实的技术基础^[1-3]。目前用电设备管理系统大多是对设备信息的静态管理, 实现设备基本信息的添加、删除、记录等管理功能, 对用电设备工作状态实时管理的系统却不多见^[4-6]。工作电流是用电设备重要的运行数据, 通过电流能反映设备开关机状况和运行状况, 可准确记录设备利用率等^[7]。本文提出并设计了一种基于电流检测的用电设备运行状态监控系统。

1 系统方案设计

系统由采集节点、路由、云服务器等组成, 整体结构框图如图 1 所示。通过在用电设备的电源侧安装电流采集

节点, 获取设备工作电流, 通过 WiFi 收发模块传输到云服务器。考虑到 WiFi 和云服务器之间通讯距离可能较远 (超出 WiFi 的最大传输范围), 因此以路由作为桥梁, 数据经过互联网实现跨地域远距离传输。云服务器中部署的监控系统获取到采集节点数据时, 根据预设的监控策略进行实时显示或者报警, 并将处理后的数据保存至数据库, 以便日后查询和统计等。系统采用 B/S 架构网络管理模式, 支持多用户机制。用户可以通过计算机、手机中的浏览器, 在任何时间, 不受地域限制地对用电设备的实时运行状态进行查看。

2 采集节点设计

采集节点选用意法半导体公司生产的 STM32F411 芯片作为主控芯片, 该微处理器具有功耗低, 工作稳定等特点^[8], 能满足整个系统的需求。系统中使用 USART1 获取设备电流值, USART2 打印参数信息, 方便查看调试信息, USART6 与 WiFi 收发模块进行数据通讯。硬件逻辑如图 2 所示。外围电路包括电源稳压模块, 电流采集模块, WiFi 收发模块等。

2.1 电流采集模块

电流采集模块的设计是本文的重点, 是获取设备电流

收稿日期: 2018-07-09; 修回日期: 2018-08-09。

基金项目: 浙江省科技计划项目(2017C33149)。

作者简介: 周 晓(1971-), 男, 浙江永康人, 副教授, 硕士生导师, 主要从事嵌入式系统、无线传感网络方向的研究。

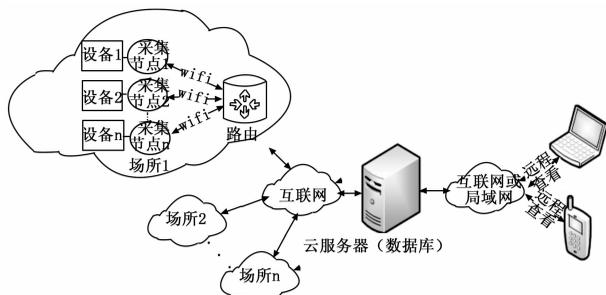


图 1 系统整体结构框图

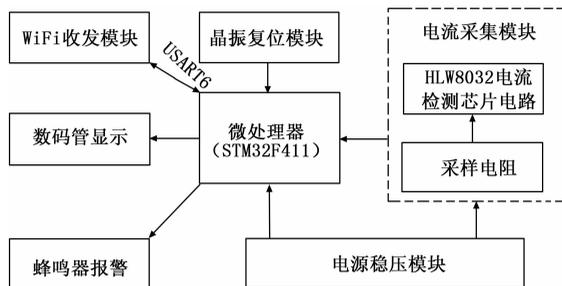


图 2 采集节点硬件逻辑框图

值精确的关键, 主要由 HLW8032 电流检测芯片和采样电阻等构成, 电路原理图如图 3 所示。HLW8032 是深圳合力为科技有限公司推出的单相功率计量芯片, 是一款高精度的电能计量 IC, 内置了晶振和参考电源, 不需使用复杂的设计电路和编写复杂的软件, 具有外围电路简单、体积小、价格便宜的优点。在 1000:1 的动态范围内有效电流的测量误差仅仅为 0.5%^[9]。采样电阻使用锰铜电阻 (成本低、精度高、温度特性好) 采样设备电流, 将获取的电流信号通过芯片内部的 ADC 以及串口的转换, 实现将电流值模拟量转化为数字量输出。

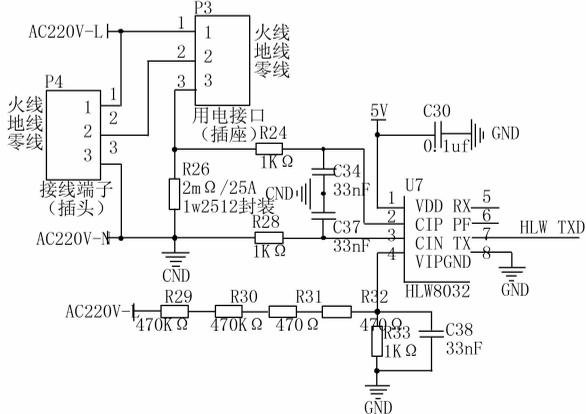


图 3 电流采集模块原理图

HLW8032 芯片引脚 1 接电源稳压模块提供的 5 V 工作电压, 增加的电容主要起滤波作用。为防止出现电压浮空而烧坏芯片的现象, 将电路中的 GND 和 AC220V-N 作为同一个参考点。根据手册最大采样电流=最大压差/采样电阻阻值可知, 当芯片引脚 2 和 3 承受 39 mV 最大压差时,

能测量设备最大电流为 19.8 A, 如果设备的电流超过此测量范围, 可以通过改变采集模块的采用电阻大小来测量更大电流的用电设备。引脚 4 实现用电设备电压采集, 采用连续 4 个相同电阻来消除单个电阻耐压不足和电网电压波动问题, 并通过由 R33 和 C38 组成的 RC 网络进行滤波。引脚 7 为串口发送, 实现和 STM32F411 中 USART1 接收引脚之间数据通讯。

2.2 微处理器

STM32F411 与 HLW8032 以 4800bps (波特率) 通讯, 并获取存放在 HLW8032 芯片寄存器的电流数据, 流程图如图 4 所示。当程序进入 USART1 中断, 通过对接收的数据进行判断, 当连续四个字节数据依次为 0x5A, 0x02, 0xD8, 0x20 时, 判定为接收到正确的电流数据, 根据 HLW8032 芯片手册可知需一次性连续接收 24 字节数据, 将接收到的这些数据经过函数处理。在处理函数中判断 HLW8032 芯片中状态寄存器的数据大小, 当状态寄存器为 0x55 时, 此时芯片误差修正功能正常, 电流参数寄存器可用, 且电流寄存器未溢出, 根据数据更新状态寄存器状态得出电流系数和电流周期, 最终根据电流值的计算公式得到设备的电流值。

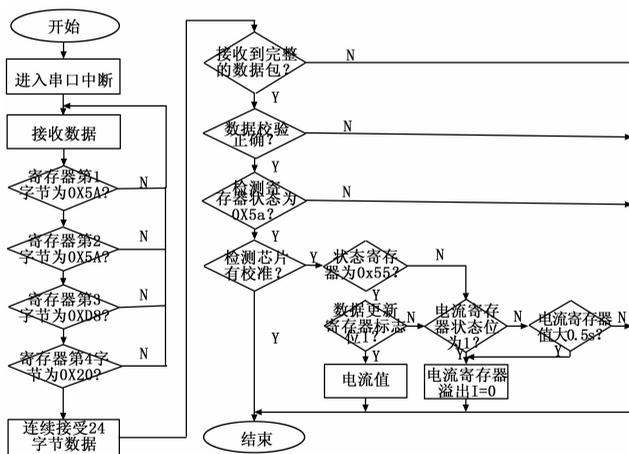


图 4 微处理器获取设备电流值流程图

电流值的计算方式为 $I = (\text{电流寄存器参数寄存器} / \text{电流寄存器}) * \text{电流系数}$ 。由于选用 2 毫欧锰铜电阻采集设备电流, $R = 0.002 \Omega$, 则电流系数 $= 1 / (R * 1000) = 0.5$ 。

电流数据经由 WiFi 收发模块上传到云服务器前需要对数据进行处理。考虑到设备启动电流有可能几倍于工作电流, 故从设备稳定工作后开始采集电流数据。将得到的电流数据以三种不同频率上传: 当电流值在额定电流值的一定范围内 (1.2 倍) 内波动时, 数据以一定频率 (0.1 Hz) 上传; 当电流值在额定电流值的一定范围内 (1.2 倍~2 倍) 内波动时, 数据以较快频率 (0.2 Hz) 上传; 当电流值超过额定电流值 (2 倍) 时, 数据以更快频率 (1 Hz) 上传。处理电流数据策略如图 5 所示。

2.3 WIFI 收发模块

WiFi 收发模块是连接各个电流采集模块与云服务器的

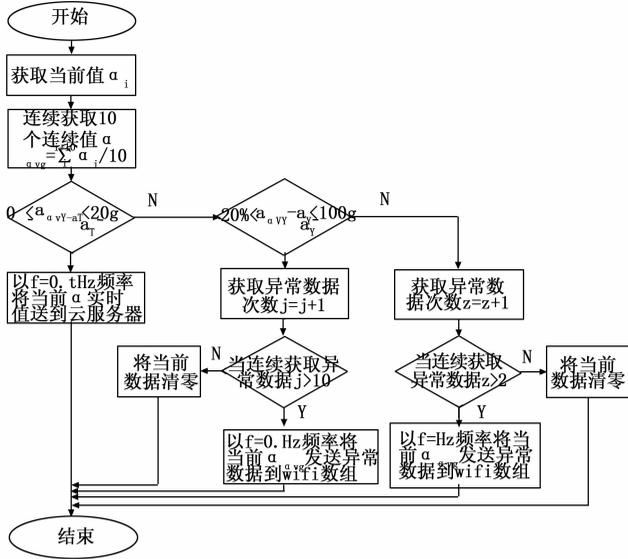


图 5 微处理器处理电流数据策略流程图

纽带。相比 zigbee 组网复杂，成本较高，传输速率低等不足，本系统采用 ESP8266 型号的 WiFi 收发模块对微处理器处理的设备数据进行获取，同时和云服务的通讯也通过该模块。ESP8266 是乐鑫公司的一款 WiFi 产品，其价格低廉，具有高性价比，是目前市面上非常流行的一款 WiFi 芯片。其具有不同工作模式，且支持透明传输。本设计采用透明传输模式（数据不发生任何形式的改变，传输过程透明）传输数据，为使 WiFi 收发模块与云服务器远距离通讯，需进行联网配置，传统的配网方式中采集节点连接新路由时需要修改底层代码中路由的 SSID 和 PSW，本系统使用网页配网方式将 SSID 和 PSW 动态存入 EEPROM 中使设备数据经过路由接入云服务器。网页配网数据传输流程如图 6 所示。

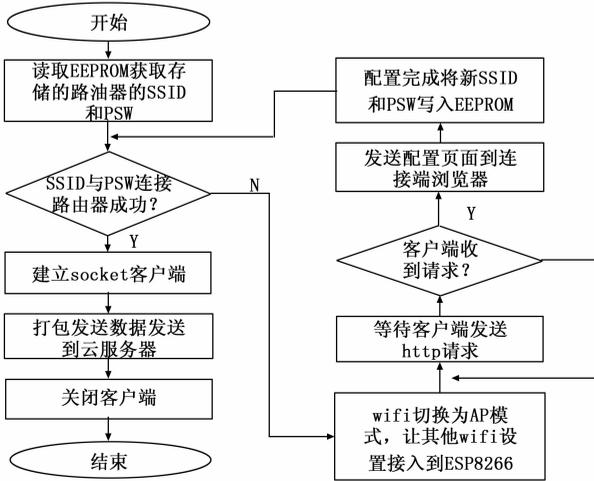


图 6 网页配网数据传输流程图

WiFi 收发模块读取到正确的 SSID 和 PSW 连接路由成功后，需要对其进行如下配置：WiFi 模块重启操作，AT+

RST；配置数据加密方式，网络访问密码，AT + WSKEY = WPA2PSK，配置 ESP8266 当前 WiFi 模式为 station 模式，AT+CWMODE=STA，STA 组网方式是系统中由一个路由器作为无线网络的中心节点，采集节点作为无线站点的组网方式；设置 WiFi 模块为单连接模式，AT+CIPMUX=DISABLE；创建 TCP 连接（Socket 客户端），配置云服务器名称（IP 地址），80 端口和第 5 通道进行数据传输，AT + CIPSTART = "TCP", "47.106.204.25", "80", "5"；在透传模式下，发送设备电流数 AT+CIPSEND = ">"，进入透传模式发送数据，每包数据以 20 ms 间隔区分。Socket 客户端配置云服务器名称的目的在于 socket 客户端对云服务器的寻址，云服务器打开 80 端口则是客户端和服务器端数据交互通道，无需通过防火墙。

WiFi 收发模块与云服务器之间的通讯方式为 RestFul API。其基于 HTTP 协议和 json 数据格式，适合平台资源管理，平台与平台之间数据对接，使用短连接上报终端数据及时间序列化数据存储等场景。其 RestFul API 应用程序设计的提出，改善了用户接口跨多个平台的可移植性，通过分离用户接口和数据存储，使得不同用户终端与单一服务器数据交互成为了可能。方案中 WiFi 收发模块作为 TCP 客户端（socket）以 RestFul API 请求方式实现与云服务器数据交互。

3 云服务器

文献 [10] 中设备运行状态监测系统以智能网关为中心，以一定场所为单点，采用局域网的形式和服务器进行数据交互，用户无法实现外网远程访问服务器中数据；本设计采用局域网与互联网相结合的方式，以云服务器为中心，进行数据接收和处理，从而实现外网远程访问。云服务器采用腾讯云服务器，是一种处理性能优越、安全可靠的计算服务，方便用户建立数据中心，可实现数据的灵活访问和数据存储计算^[11]。云服务器的系统配置如表 1 所示。

表 1 系统配置

配置	云服务器
处理器	Intel Xeon 单核 CPU
内存容量	DDR4 2 GB(2133MHz)
硬盘	40 GB SSD 盘
操作系统	Windows 2012 R2 标准版 64 位中文版

云服务器分为控制台、后台服务和数据库三部分，控制台的功能相当于云服务器与外部应用进行沟通的桥梁，控制台向下与采集节点通讯，采集节点中 WiFi 收发模块作为 socket 客户端向云服务器的控制台传输设备电流数据。数据库服务器主要是存储采集的电流数据，以使用户通过网页随时进行查询设备运行状态。WEB 服务器端程序部署在 Tomcat 容器中，用 Java 语言编写，采用 MVC 分层设计模型。数据库采用 Mysql 数据库。

在 WEB 服务器端程序设计中，为了同时接收多个用电

设备的数据, 使用 Java 多线程的方式来处理 socket 客户端传过来的 json 格式数据 (key-value 键值对), 具体流程如下: 服务器根据 socket 类型创建 socket, 服务器为 socket 绑定对应的 IP 地址和端口号 (80), 服务器接收到了用户发来的 socket 连接请求时, 被动打开 socket, 开始接收客户端请求, 直到 socket 客户端返回连接信息。这时候服务器 socket 进入堵塞状态, 接收客户端连接信息后返回, 然后开始接收下一个用户端请求。服务器端启动 `socket.getInputStream()` 与客户端建立连接, 获取 WiFi 收发模块输入的 json 数据, 将其转化为数组类型, 调用 `socket.substring()` 方法解析数据, 将这些数据经过加载 mysql 的 jdbc 驱动, 建立与 mysql 数据库连接, 执行 `insert` 语句后存入数据库进行长期保存。

云服务器主要分 4 个功能: 用户管理、实时运行状态、电流数据查询、日志记录, 如图 7 所示。

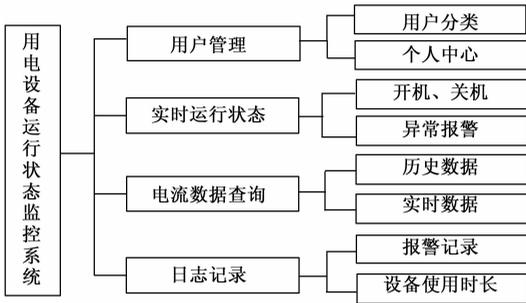


图 7 云服务器功能结构示意图

云服务器使用 `XMLHttpRequest` 对象异步读取数据库中数据, 采用异步 `AJAX` 技术 (不重新加载整个页面的情况下, 可以与服务器交换数据并更新部分网页内容) 对获取的数据进行页面实时显示。

4 实验测试分析

4.1 设备电流精度分析

以实验室的电风扇、电脑、电热壶作为实验对象。实验步骤: 采集节点上电后, 微处理器初始化, WiFi 模块切换为 STA 模式, 接收网页发送连接路由的 SSID 和 PSW, 和云服务器建立连接, 将采集节点获取的此三种设备电流值传输到云服务器上。如图 8 所示, 通过 `USART2` 查看具体连接协议和电流值 (此为电风扇第一次传输数据)。如图 9 所示, 通过协议实时将设备数据经由网络写入到云服务器

数据库 (Mysql) 中, 传输协议中前四位表示的是设备的电流值, 第五位表示的是设备的当前状态, 第六位和第七位分别表示的是设备号和实验室号。

对不同设备电流的实时检测, 选取不同次数的电流采集模块测量值和电流表测量值进行比较计算, 如表 2 所示误差在 2% 以内。

表 2 实验测量结果

实验设备/额定电流 选取次数		电风扇 (0.159A)	电脑 (0.327A)	电热壶 (8.182A)
		第 1 次数据	电流采集模块测量/A 0.152	0.321
	电流表测量值/A 0.150	0.319	8.168	
	相对误差/% 1.333	0.940	0.183	
第 100 次数据	电流采集模块测量/A 0.148	0.327	8.171	
	电流表测量值/A 0.146	0.321	8.156	
	相对误差/% 1.369	1.869	0.184	
第 200 次数据	电流采集模块测量/A 0.147	0.329	8.179	
	电流表测量值/A 0.146	0.324	8.156	
	相对误差/% 0.685	1.543	0.282	

4.2 设备运行状态测试

为模拟微处理器处理电流数据策略的准确性, 以实验室 (研究室) 的 4 台相同功率的电风扇作为实验对象, 模拟设备正常工作, 不同报警的状态。在实验时, 设定电风扇的阈值 (定于 0.13 A), 开启电风扇 1 处于正常档 (一档), 电风扇 2 处于较大风力档 (二档), 电风扇 3 处于更大风力档 (三档), 电风扇 4 处于关机。

通过实验室用电设备运行状态监测系统的界面可以查看 4 台电风扇的运行状态, 绿色表示设备正常, 灰色表示设备关机, 黄色 (一级报警) 表示设备电流值超过设定阈值的 20%, 红色 (二级报警) 表示设备电流值超过设定阈值的两倍。如图 10 所示。

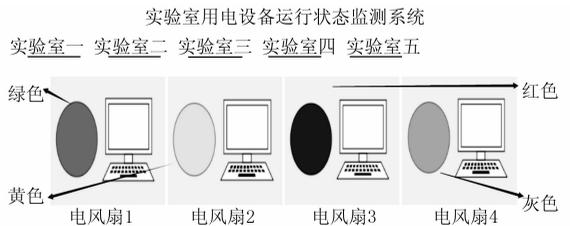


图 10 电风扇运行状态的实时监控界面

根据设备运行状态, 查看数据库中实时的运行信息, 如图 11 所示。其中 `eid` 字段为设备 id 号, `ename` 为设备的名称, `elstarttime` 为设备开始工作的时间, `elendtime` 为设备停止工作的时间, `eltime` 为设备从开始运行到停止运行所经历的时长, `lid` 为设备所处实验室 id 号, `lname` 为设备所处实验室名称, `elvalue` 为设备关机之前最新一次电流值, `estate` 为设备当前所处的状态 (0 表示设备当前状态为正常, 1 表示一级报警, 2 表示二级报警, 3 表示关机)。根据设定电风扇电流值阈值, 获取运行状态, 电风扇在数据库中



图 8 WiFi 连接协议图

图 9 传输协议图