

文章编号: 2095-2163(2022)01-0099-05

中图分类号: TP391

文献标志码: A

基于 Arduino 的机房温湿度监控系统设计与实现

李 博, 杨 俊, 王中行, 于海鹏

(河南工程学院 计算机学院, 郑州 451191)

摘 要: 本文利用计算机通信、数据库、传感器技术等,设计了一套机房温度监控系统,实现了对机房内温湿度的有效监控。设计以 arduino 开发板、dht11 温湿度传感器以及远程监控系统为核心,通过温湿度模块来检测机房温度是否超出正常标准,如果超标自动打开风扇进行散热。温湿度经 wifi 模块发射到指定服务器,通过监控系统进行实时检测。温湿度传感器灵敏度可根据机房环境进行灵活调节,通过借助电信网络,使传输便捷,从而解决目前温度监控中存在的一些弊端。人性化的操作系统界面,为管理人员提供实时、准确、有效的机房环境数据,预警和自处理功能,保障了机房的安全稳定运行。

关键词: 物联网; 机房监控; B/S 架构; 远程监控

Design and implementation of temperature monitoring system in computer room based on Arduino

LI Bo, YANG Jun, WANG Zhonghang, YU Haipeng

(College of Computer Science, Henan University of Engineering, Zhengzhou 451191, China)

【Abstract】 In this paper, the use of computer communication, database, sensor technology, design a computer room temperature monitoring system, to achieve effective monitoring of temperature and humidity in the computer room. The design takes The Arduino development board, DHT11 temperature and humidity sensor and remote monitoring system as the core. The temperature and humidity module detect whether the temperature in the equipment room exceeds the normal standard. If the temperature exceeds the standard, the fan is automatically turned on for heat dissipation. The temperature and humidity are transmitted to the specified server through the wifi module for real-time detection through the monitoring system. The sensitivity of the temperature and humidity sensor can be adjusted flexibly according to the environment of the machine room. By using the telecommunication network, the transmission is convenient, so as to solve some disadvantages existing in the current temperature monitoring. The humanized operating system interface provides real-time, accurate and effective environment data, warning and self-processing functions for the management personnel, ensuring the safe and stable operation of the computer room.

【Key words】 internet of things; computer room monitoring; B/S architecture; remote monitoring

0 引言

随着人工智能、虚拟现实等新兴技术的快速发展,互联网通信产业也在迅猛发展。伴随着网络规模的上升,机房作为互联网的神经中枢,其数量与规模也在不断朝多而广的方向发展^[1]。目前,对于银行、公安、消防等部门,机房管理存在的潜在问题尤为突出。如果因机房异常情况而导致宕机的发生^[2],将对管理机构产生巨大的负面影响。合适的温湿度环境,是中心机房设备良好运行的基础,一旦机房环境出现问题,必然会影响到计算机系统的运行,对数据的可靠传输及存储造成一定影响,就可能对机房设备造成不可恢复的损坏,造成不可挽回的损失。

目前,环境监控系统主要用于检测大面积户外环境,由于户外环境的条件恶劣,造成大部分环境监控系统结构复杂,造价高昂。室内环境一般情况属于特定环境,假使将市面上的环境监控系统照搬进室内,会造成系统的不实用性与冗余性^[3]。

物联网技术的发展为实现机房温湿度监控提供了解决条件,物联网通过有线通信或者无线通信等传输方式,将终端与互联网相连,实现数据的交换和通信,从而实现智能管理^[4]。

机房温湿度检控系统通过运用物联网技术达到实时监控机房温湿度环境变量的功能,这也是设计并实现本系统的主要目的。在机房的温湿度管理终端,对远程机房温湿度进行实时监测^[5],对可能出现的温湿度异常进行有效计算处理与提供预警,将

基金项目: 河南省高等学校重点科研项目(20A520010)。

作者简介: 李 博(1982-),男,博士,讲师,主要研究方向:物联网应用;杨 俊(2001-),男,本科生,主要研究方向:智能物联;王中行(2000-),男,本科生,主要研究方向:大数据处理;于海鹏(1976-),男,硕士,副教授,主要研究方向:计算机应用。

收稿日期: 2021-09-22

可能影响机房运转的异常处理在萌芽之中,保证机房的正常运转,同时减轻机房工作人员的负担,提高机房的利用率,实现机房的智能管理。

1 国内外研究现状

1999年物联网的概念首次被提出,2005年国际电信联盟首次对物联网进行阐述,定义物联网为通过信息采集设备,实时采集用户所需信息^[6],并与互联网结合为一个巨大的网络。其目的是实现人与物、物与物的连接。物联网概念一经提出,就被广泛应用^[7]。如在智能交通,智能家具,公共安全,智能汽车,生物医疗等领域的应用,给人类生活带来巨大便利。

2004年日本提出自己的物联网国家信息化战略,通过物联网促进技术革命,实现国民便利。基于物联网技术,松下提出了智能家电网络系统,可以实现对家庭电器网络化管理。同时,美国也将物联网技术升级为本国的国家战略,将其作为提升本国经

济的重要引擎。例如,德克萨斯推出了智能数字电网^[8],借用此系统,实现对电网异常的感知与异常处理。欧盟相应的也提出了物联网方案,在医疗健康领域实现对药品的监控,保障用户的用药安全^[9]。

国内物联网在外部环境的推动下,研究环境也在不断改善^[10]。在感知中国国家战略的引导下,物联网产业如雨后春笋,具有旺盛的生命力。研究工作着重点在物联网领域的传感器,网络协同以及软硬件结合,已经形成了一条从研发到市场应用的物联网产业链条^[11]。

2 系统结构

系统总体由传输控制系统,WEB软件层,与云端平台组成。传输控制系统负责数据的采集,通过传输介质进行传输、异常处理、数据存储;WEB软件层负责将数据库中数据采集出来供前端展示与后台处理;云平台负责数据远程存储与查询。系统总体结构如图1所示。

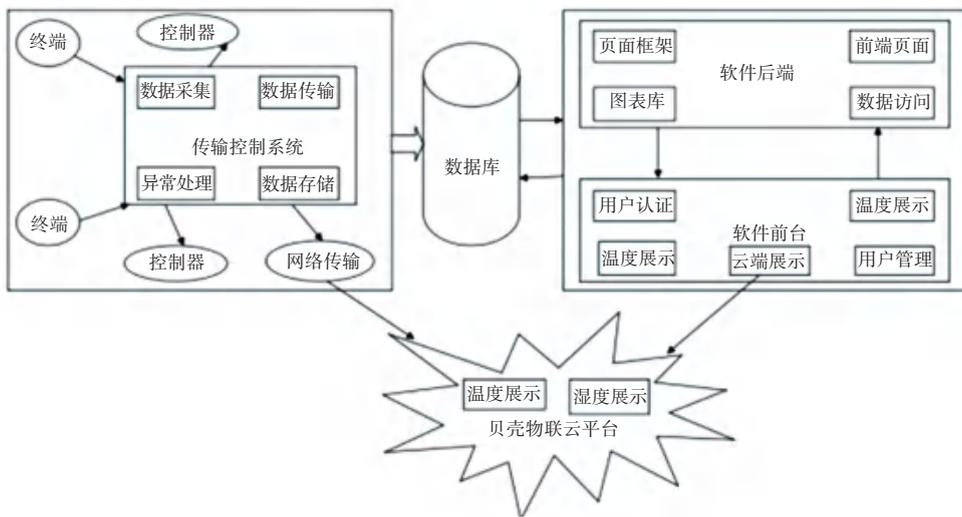


图1 系统总体结构

Fig. 1 Overall system structure diagram

2.1 底层传输控制系统

底层传输控制系统采集并处理终端获取到的机房环境数据,完成数据库中环境变量的实时更新,异常检测任务。按功能划分为:机房温湿度数据采集模块、机房温湿度数据处理模块、机房温湿度数据存储模块。总体结构如图2所示。

2.1.1 数据采集

数据采集使用传感器,一方面完成对机房环境数据的准确检测,同时将采集到的数据传送到数据通信过程中。整个过程要保证机房环境数据的准确和及时的传送。



图2 底层传输控制系统总体结构

Fig. 2 Overall structure diagram of bottom transmission control system

数据采集使用 DHT11 温湿度模块,将该模块与 Arduino 开发板进行正确连接,编写数据采集代码,

对采集模块代码进行编译。验证获得温湿度数值是否正确。数据采集源码如下:

```
#include<DHT11.h> //导入所需库文件
#define DHT11PIN 6
DHT11 DHT11; //实例化对象
void setup( )
{
Serial.begin( 152000); //设置波特率参数
pinMode( DHT11PIN, OUTPUT);
void loop( )
{
//循环
int shuju=DHT11.read( DHT11 PIN); //将读取到的值赋给 chk
int wendushuju=(float)DHT11.temperature; //将温度值赋值给 wendushuju
int shidushuju=(float)DHT11.humidity;
Serial.print( wendushuju); //打印温度结果
Serial.print( " ");
Serial.println( shidushuju);
delay( 2000); //设定延迟时间
}
```

2.1.2 数据通信

数据通信模块需要保持与传感器终端的连接,及时完成数据传送功能,并具有对数据进行简单排错功能,以减少错误数据的上传。

受工作环境的影响,为了保证系统的稳定性,采用有线通信方式,通信协议采用 Serial 串口通信进行数据传输。该通信方式稳定性高,受周围环境变化的影响范围小。由于 Python 的扩展库中包含串口通信协议 pySerial,因此在 PyCharm 中使用 pyserial 包进行代码编译,达到与 Arduino 串口通信的目的。代码实现如下:

```
import serial
s=serial.Serial( 'com4', 9600, timeout=2) #选择串口号,波特率
while True; #无限循环读取数据
n=s.readline( ) #获得端口数据
data_pre=str( n) #转化为字符串格式
tem=data_pre[ 2:4] #取字符串部分数据,温度
data2=data_pre[ 5:-5] #取字符串部分数据,湿度
times=local_time[ 11:] #取时间的部分数据
app.run( debug=True)
```

2.1.3 数据处理及异常处理

首先,数据处理模块需要对传送过来的原始数据进行校验,排出无关数据;其次,通过一定的规则,对数据进行解析,根据解析结果传输到不同的数据库中。底层传输控制系统感知终端获取到的机房环境数据,判断是否异常,根据判断结果,自动做出处理。

在 Arduino ide 编写通过 DHT11 获取温湿度的代码,打开串口调试工具,观察串口调试界面是否获

取到温湿度,再次编译代码,添加自动控制函数;根据机房最佳温度为 40°以下,因此将判断值设定为 40。当机房环境变量中的温度值大于 40 时,异常处理模块打开,对机房进行降温处理,待机房温度下降到 30°之下,风扇自动关闭。代码实现如下:

```
#include<DHT11.h> //导入 DHT11 库文件
#define DHT11PIN 12 //定义温湿度针脚号为 12
DHT11 DHT11; //实例化一个对象
const int LEDDENG=4;
void setup( ) { //设置
Serial.begin( 15200); //设置波特率参数
pinMode( DHT11PIN, OUTPUT); //设置波特率参数
pinMode( DHT11PIN, LEDDENG);
}
void loop( ) { //循环
int shuju=DHT11.read( DHT11PIN); //获取数据
int wendushuju=(float)DHT11.temperature; //获取温度
if( wendushuju>30)
{
digitalWrite( LEDDENG, HIGH);
}
else
{
digitalWrite( LEDDENG, LOW);
}
int SHIDUSHUJU=(float)DHT11.humidity;
//Serial.print( " 温度"); //打印出 Temperature;
Serial.print( WENDUSHUJU); //打印温度结果
Serial.print( " ");
Serial.println( SHIDUSHUJU);
delay( 1000);
}
```

2.2 WEB 控制系统

2.2.1 用户认证

由于 HTTP 其安全性较低,而本 WEB 软件需要对使用者进行身份认证以及跟踪操作。因此,当用户登录网站时,WEB 软件记录用户信息,匹配后台数据库,根据匹配结果,给用户设定操作权限。该模块还提供用户注册界面,给未注册的用户注册权限。

WEB 用户认证过程中需要用到 Flask—Login 与 Werkzeug 扩展模块;此外,还需要 Flask—bootstrap 前端引擎渲染 HTML 模板文件。

用户使用账号、密码进行登录时,若用户名及密码与后台匹配均正确,跳转到 home 页面,否则返回登录界面,进行再次登录。代码实现如下所示:

```
deflogin( ): if request.method == 'GET':
return render_template( 'denglu.html')
else:
username=request.form.get( ' yonghuming')
```

```
password1 = request.form.get('mima1')
xinxi = validate(yonghuming, mimal)
if '数据存在数据控库' in message:
    session['yonghuming'] = username
    session.permanent = True
    return redirect(url_for('zhuye'))
else:
    flash(message)
    return render_template('denglu.html')
```

2.2.2 数据库访问与前端页面可视化

本系统使用 MySQL 数据库存储信息,使用 python 语言和 pyMySQL 完成系统与 MySQL 数据库的通信。将对象写入数据库,Flask—SQLAlchemy 为每个模型类都提供了 query 对象,可以进行数据库查询。代码实现如下所示:

```
new_user = Users(username = username, password = password1)
db.session.add(new_user)
db.session.commit()
```

通过 link 标签引入 jquery.min.js、bootstrap.min.js 文件,使一个常规的 H5 页面转化为 bootstrap 模板,为 HTML 页面标签中的 class 属性添加 bootstrap 前端框架含有的属性,实现实例化 Bootstrap 的功能创建网页。

采用图表曲线的方式展示机房环境变化,能够给用户直观的感受,从而使用户更加清楚准确的掌握机房环境的变化。本系统采用 Echart 图表库(使用 JavaScript 实现的开源可视化库),在 data 页面中引入 echart.js 库,实现网页中交互性图表及浏览器页面中数据图表的展示。

3 实验验证

3.1 数据传输

使用命令行工具打开数据库。在命令提示符下输入 MySQL—localhost -uroot。选择本地数据库并输入密码,使用 MySQL 查询,结果如图 3 所示。

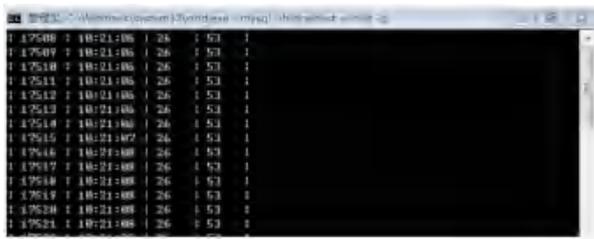


图 3 数据传输结果

Fig. 3 Experimental results of data transmission

3.2 读取机房温度数据

当传感系统正常工作时,打开机房温度监控系统进行登录操作。选择温度或湿度展示,前端页面

将展示温湿度曲线图。温湿度曲线如图 4、图 5 所示。

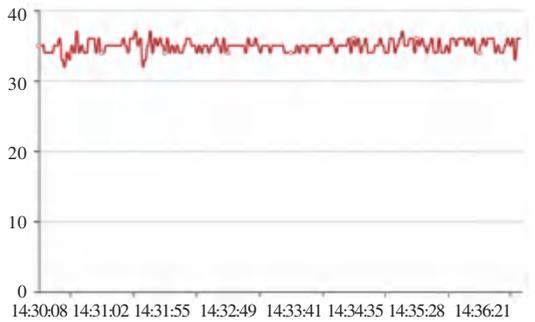


图 4 温度曲线

Fig. 4 Temperature curve

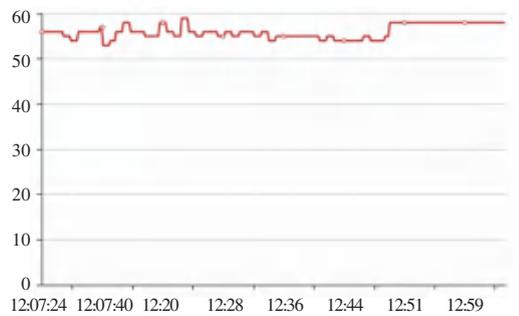


图 5 湿度曲线图

Fig. 5 Humidity curve

3.3 用户管理

若以管理员身份登录系统,选择查看用户信息,可以查看最近登录用户信息,其中包括用户 ID、用户名、用户密码和创建时间等,同时软件系统支持对用户信息进行管理。为及时发现底层传感设备的情况,系统支持用户通过 Web 页面对监控系统进行反馈,实现用户与系统管理员的异步交互。用户反馈数据展示如图 6 所示。

序号	标题	内容
1	温度传感器	不灵敏
2	温度传感器	不灵敏
3	温度传感器	不灵敏
4	温度传感器	不灵敏
5	温度传感器	不灵敏
6	温度传感器	不灵敏
7	温度传感器	不灵敏
8	温度传感器	不灵敏
9	温度传感器	不灵敏
10	建议	期待下一个版本
11	建议	功能太少

图 6 用户反馈数据展示

Fig. 6 User feedback data display diagram

3.4 云端存储

真实环境下,不仅要求数据可以在局域网络查

看,还应当支持远程广域网查看。本系统连接物联网云平台,对用户数据的云端实时查询服务。通过系统的 WIFI 模块,向云端服务器发送温湿度

数据,将数据存储在云端。用户通过登录云平台,访问本地机房温湿度的实时数据。从云端访问本地机房温湿度的数据曲线如图 7、图 8 所示。

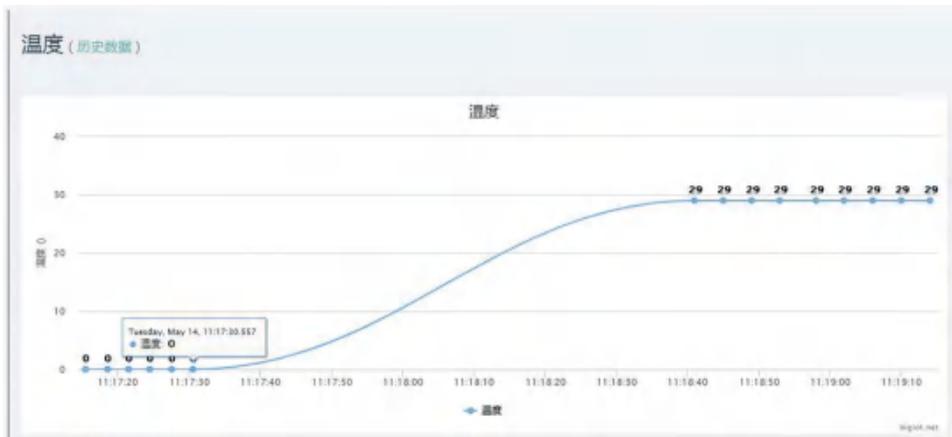


图 7 温度展示图

Fig. 7 Temperature display diagram

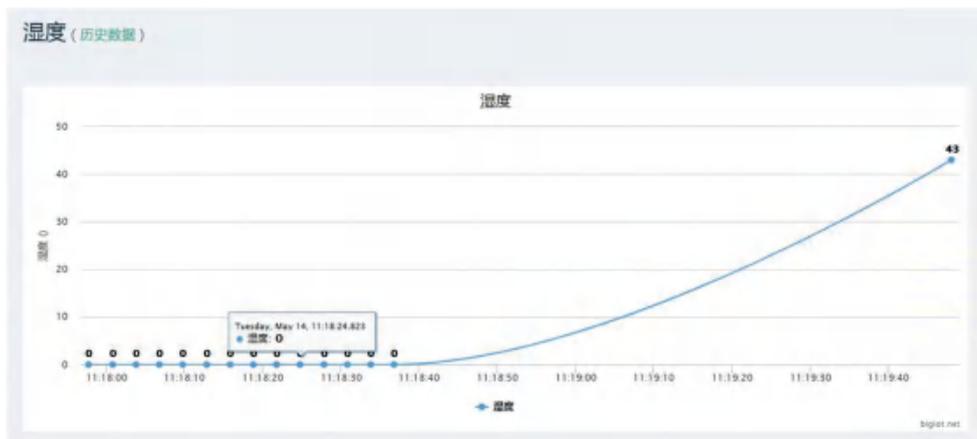


图 8 湿度展示图

Fig. 8 Humidity display diagram

4 结束语

本研究在了解了国内机房温度设备远程监控的现状后,针对机房监控存在的突出问题,设计了有别于传统监控软件的监控系统。与现存的环境监控系统相比,本系统具有稳定性好、成本低廉的优势。

尽管本文完成了机房监控系统的设计和实现,但由于知识水平、实践经验等限制,在许多方面还不够完善,还有许多不足需要解决。接下来,对数据展示功能继续更新,给用户提供更好的使用效果;针对用户身份的不同,保证数据的安全使用,为用户分配合理权限;接入云平台的邮件通知,为基于 Arduino 的机房温湿度监控系统产品提供预警功能。

参考文献

[1] 蔡庆瑶. 基于物联网的智能安防系统设计[D]. 黑龙江:哈尔滨理工大学,2018.

- [2] 郭庆飞. 基于 Android 的物联网智能家居远程控制系统设计[D]. 山东:山东工商学院,2018.
- [3] 王鑫. 基于物联网的温室监测管理系统设计[D]. 湖南:湖南大学,2018.
- [4] 张庆雷. 基于物联网架构的可移动温室环境监测系统研究[D]. 山东:山东农业大学,2018.
- [5] 崔婷婷. 基于 Flask 框架的骆驼帮系统的设计与实现[D]. 山东:山东大学,2016.
- [6] 李磊. 基于 WEB 软件的分布式网络实时监控系统的设计与实现[D]. 山东:山东大学,2016.
- [7] 刘厉. 智能家居服务系统的设计与实现[D]. 江苏:东南大学,2017.
- [8] 杨劲柯. 基于物联网的仓储搬运设备远程监控系统软件设计与实现[D]. 江苏:东南大学,2017.
- [9] 莫佳荣. 基于物联网的温湿度测量系统的研制[D]. 广东:广东工业大学,2016.
- [10] 苏亮亮. 基于无线物联网的智能家居系统设计[D]. 安徽:安徽理工大学,2018.
- [11] 孙忠祥. 基于设备云平台的智能农业温室大棚远程监控系统的实现[D]. 黑龙江:哈尔滨理工大学,2017.