

毕业设计（论文）检测系统

文本复制检测报告单(全文标明引文)

№:BC202205241708004766194299

检测时间:2022-05-24 17:08:00

篇名: 基于物联网的温度控制系统设计

作者: 黄锦灿(2018211616)

指导教师: 朱敏(副教授)

检测机构: 合肥工业大学

提交论文IP: 36.***.***.***

文件名: 毕业论文定稿.docx

检测系统: 毕业设计（论文）检测系统（毕业设计（论文）管理系统）

检测类型: 毕业设计论文

检测范围: 中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

源代码库

CNKI大成编客-原创作品库

时间范围: 1900-01-01至2022-05-24

检测结果

去除本人文献复制比: 2.9%

去除引用文献复制比: 2.9%

单篇最大文字复制比: 0.6% (从“盆景”走向“风景”)

跨语言检测结果: -

总文字复制比: 2.9%

重复字数: [1085]

总段落数: [4]

总字数: [37987]

疑似段落数: [4]

单篇最大重复字数: [240]

前部重合字数: [579]

疑似段落最大重合字数: [579]

后部重合字数: [506]

疑似段落最小重合字数: [75]



指标: ☐ 疑似剽窃观点 ☒ 疑似剽窃文字表述 ☐ 疑似整体剽窃 ☐ 过度引用

相似表格: 2

相似公式: 0

疑似文字的图片: 0

5.9%(579) 5.9%(579) 基于物联网的温度控制系统设计_第1部分 (总9766字)

0.8%(75) 0.8%(75) 基于物联网的温度控制系统设计_第2部分 (总9946字)

2.2%(212) 2.2%(212) 基于物联网的温度控制系统设计_第3部分 (总9685字)

2.5%(219) 2.5%(219) 基于物联网的温度控制系统设计_第4部分 (总8590字)



(注释: 无问题部分 文字复制部分 引用部分)

1. 基于物联网的温度控制系统设计_第1部分		总字数: 9766
相似文献列表		
去除本人文献复制比: 5.9%(579) 文字复制比: 5.9%(579) 疑似剽窃观点: (0)		
1	<u>从“盆景”走向“风景”</u> 本报记者 乔金亮 - 《经济日报》- 2022-05-09	2.5% (240) 是否引证: 否
2	<u>“十四五”全国农业农村信息化发展规划</u> - 《中国畜牧业》- 2022-03-25	1.9% (187) 是否引证: 否
3	<u>发展智慧农业 建设数字乡村 以信息化引领驱动农业农村现代化</u> - 《中国畜牧业》- 2022-03-25	1.7% (167) 是否引证: 否
4	<u>中国共产党人对社会主义现代化的探索与追求</u> 范希春; - 《中共杭州市委党校学报》- 2022-01-10	0.9% (89) 是否引证: 否
5	<u>基于Zigbee/GPRS物联网网关系统的设计与实现</u> 赵玉超; - 《长江信息通信》- 2021-12-15	0.8% (83) 是否引证: 否
6	<u>乡村振兴与乡村现代化</u> - 《人民论坛》- 2022-01-05	0.8% (75) 是否引证: 否
7	<u>习近平新时代观的生成逻辑探析</u> 程权杰; - 《漳州职业技术学院学报》- 2021-12-15	0.6% (59) 是否引证: 否
8	<u>关于全面推进乡村振兴的对策思考</u> 汪洋; - 《中国市场》- 2022-01-08	0.6% (54) 是否引证: 否
9	<u>巩固拓展脱贫攻坚成果同乡村振兴有效衔接的长效机制与政策研究</u> 尹成杰; - 《华中师范大学学报(人文社会科学版)》- 2022-01-21	0.6% (54) 是否引证: 否
10	<u>农村物流助推农业农村现代化发展研究——以辽宁省朝阳市喀左县为例</u> 冯志强;吴昊;李福宝;谢宝玲; - 《北京农业职业学院学报》- 2022-01-20	0.6% (54) 是否引证: 否
11	<u>日照市: 在乡村振兴中彰显人大力量</u> 赵振霖; - 《山东人大工作》- 2021-12-25	0.6% (54) 是否引证: 否
12	<u>习近平法治思想指引下的乡村振兴法治保障</u> 孟俊涛;朱振辉; - 《西藏发展论坛》- 2022-02-08	0.6% (54) 是否引证: 否
13	<u>践初心使命 解“急难愁盼”</u> 郭静; - 《内蒙古人大》- 2021-09-10	0.6% (54) 是否引证: 否
14	<u>弘扬脱贫攻坚精神 全面推进乡村振兴</u> - 《记者观察》- 2021-11-25	0.6% (54) 是否引证: 否
15	<u>新时代乡村教育振兴的新目标与新路径</u> 郝文武; - 《陕西师范大学学报(哲学社会科学版)》- 2022-01-15	0.6% (54) 是否引证: 否
16	<u>高等教育助力乡村振兴的时代诠释</u> 于东超; - 《中国高等教育》- 2021-11-18	0.6% (54) 是否引证: 否
17	<u>乡贤文化在乡村振兴中的功能释放及法治路径</u> 宋才发; - 《社会科学家》- 2021-12-05	0.6% (54) 是否引证: 否
18	<u>沟通与治理: 乡村振兴下的县级融媒体中心建设</u> 李珮;张璐璐; - 《中国编辑》- 2022-02-20	0.6% (54) 是否引证: 否
19	<u>论新发展阶段全面推进乡村振兴</u> 陈文胜;李珊珊; - 《贵州社会科学》- 2022-01-20	0.6% (54) 是否引证: 否
20	<u>动员式治理与嵌入式交往: 驻村干部工作艺术与乡村振兴</u> 谢治菊;卢荷英; - 《湖北民族大学学报(哲学社会科学版)》- 2022-03-07 19:18	0.6% (54) 是否引证: 否
21		0.6% (54)

	党建引领村级集体经济巩固脱贫成果的重要作用——以山西省静乐县杨家山村为例	
	田孔社; - 《活力》 - 2021-10-30	是否引证: 否
22	共同富裕和人的全面发展视域下的乡村振兴	0.6% (54)
	左停;刘路平; - 《贵州社会科学》 - 2022-02-20	是否引证: 否
23	《求是》杂志发表习近平总书记重要文章——《坚持把解决好“三农”问题作为全党工作重中之重, 举全党全社会之力推动乡村振兴》	0.6% (54)
	- 《小康》 - 2022-04-11	是否引证: 否
24	优化京津冀空间布局, 助力生态建设和经济发展——河北省区域科学学会成立大会暨太行论坛2021年会专家发言摘编	0.6% (54)
	刘嘉麒;杨开忠;肖金成;王留根; - 《河北地质大学学报》 - 2022-02-20	是否引证: 否
25	《求是》杂志发表习近平总书记重要文章 《坚持把解决好“三农”问题作为全党工作重中之重, 举全党全社会之力推动乡村振兴》	0.6% (54)
	- 《社会主义论坛》 - 2022-04-05	是否引证: 否
26	本期特稿	0.6% (54)
	- 《乡村振兴》 - 2022-04-15	是否引证: 否
27	坚持把解决好“三农”问题作为全党工作重中之重 举全党全社会之力推动乡村振兴	0.6% (54)
	习近平; - 《中国民政》 - 2022-04-15	是否引证: 否
28	从西柏坡到北京: 新中国成立前人民政协初心和使命的确立	0.5% (50)
	张志永; - 《河北省社会主义学院学报》 - 2022-01-15	是否引证: 否
29	农村电商发展背景下河北省农产品网络品牌建设路径研究	0.5% (45)
	何雅洁;王振杰;戎钰; - 《全国流通经济》 - 2020-12-08	是否引证: 否
30	基于ARM与Linux的无线传感器网络节点设计与实现	0.3% (29)
	周应宾(导师: 陈红) - 《西南交通大学硕士论文》 - 2009-05-01	是否引证: 否

原文内容

学号： 2018211616 密级：
Hefei University of Technology
本科毕业设计（论文）
UNDERGRADUATE THESIS

类型:	设计
题目:	基于物联网的温度控制系统
专业名称:	自动化
入校年份:	2018级
学生姓名:	黄锦灿
指导教师:	朱敏副教授
学院名称:	电气与自动化工程学院
完成时间:	2022年5月

类型： 设计
题目： 基于物联网的温度控制系统
专业名称： 自动化
入校年份： 2018级
学生姓名： 黄锦灿
指导教师： 朱敏副教授
学院名称： 电气与自动化工程学院
完成时间： 2022年5月
合肥工业大学
本科毕业设计（论文）
基于物联网的温度控制系统

学生姓名:	黄锦灿
学生学号:	2018211616
指导教师:	朱敏副教授
专业名称:	自动化
学院名称:	电气与自动化工程学院

学生姓名： 黄锦灿
学生学号： 2018211616

指导教师： 朱敏副教授
专业名称： 自动化
学院名称： 电气与自动化工程学院
2022 年 5 月
A Dissertation Submitted for the Degree of Bachelor
Temperature Control System Based on Internet of Things
By
Huang JinCan
Hefei University of Technolog

学位论文作者签名：	指导教师签名：
签名日期： 年月日	签名日期： 年月日

y
Hefei, Anhui, P.R.China
Month 5 , Year 2022
毕业设计（论文）独创性声明

本人郑重声明：所呈交的毕业设计（论文）是本人在指导教师指导下进行独立研究工作所取得的成果。据我所知，除了文中特别加以标注和致谢的内容外，设计（论文）中不包含其他人已经发表或撰写过的研究成果，也不包含为获得合肥工业大学或其他教育机构的学位或证书而使用过的材料。对本文成果做出贡献的个人和集体，本人已在设计（论文）中作了明确的说明，并表示谢意。

毕业设计（论文）中表达的观点纯属作者本人观点，与合肥工业大学无关。

毕业设计（论文）作者签名： 签名日期： 年月日

毕业设计（论文）授权使用授权书

本学位论文作者完全了解合肥工业大学有关保留、使用毕业设计（论文）的规定，即：除保密期内的涉密设计（论文）外，学校有权保存并向国家有关部门或机构送交设计（论文）的复印件和电子光盘，允许设计（论文）被查阅或借阅。本人授权合肥工业大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库，允许采用影印、缩印或扫描等复制手段保存、汇编毕业设计（论文）。

（保密的毕业设计（论文）在解密后适用本授权书）

学位论文作者签名： 指导教师签名：

签名日期： 年月日 签名日期： 年月日

摘要

现阶段，我国的农业信息化处于起步阶段，网络基础设施不完善，农业物联网设备缺乏。农业信息化可以借鉴工业控制网络，但是，两者有不同之处。应用于工业的DCS和FCS适用于较大的生产规模，而农业是以小生产规模农户为主，同时存在着大规模的企业。因此，需要一个能够适应这两种规模的物联网系统。此外，当前市面上的孵化箱采用的是基于回差温度带的开关量控制，在纯滞后较大时控制质量难以保证，因此，决定设计一个基于物联网的温度控制系统，温度控制的对象为水床孵化箱。

首先，针对孵化箱对温度的要求，设计了温度控制器的硬件，对PI，模糊PI以及两者与史密斯预估器结合的控制算法进行了仿真，并进行了对比分析。

第二，设计了一个以MQTT（Message Queuing Telemetry Transport，消息队列遥测传输）协议为核心的物联网系统，该系统能够调整拓扑结构以适应不同生产规模，所有控制器采用同一种接口，控制器不局限于温度控制器，并且能够扩展各种类型的上位机。此外设计了该物联网系统的安全认证程序。

最后，设计了网页客户端，给登录用户提供对不同类型的控制器的控制面板，以图形界面显示控制器的状态，能够对控制器下发控制命令。

关键词：农业信息化；物联网；温度控制；MQTT；网页客户端

ABSTRACT

At this stage, China's agricultural informatization is in its infancy, with imperfect network infrastructure and lack of Agricultural Internet of things equipment. Agricultural informatization can learn from industrial control network, but they are different. DCS and FCS applied to industry are suitable for large-scale production, while agriculture is dominated by small-scale farmers, and there are large-scale enterprises at the same time. Therefore, we need an Internet of things system that can adapt to these two scales. In addition, the current incubator on the market adopts the switching value control based on the return difference temperature band, which is difficult to ensure the control quality when the pure lag is large. Therefore, it is decided to design a temperature control system based on the Internet of things, and the object of temperature control is the water bed incubator.

Firstly, according to the temperature requirements of the incubator, the hardware of the temperature controller is designed. The PI, fuzzy PI and the control algorithm combined with Smith predictor are simulated and compared.

Secondly, an Internet of things system with mqtt (message queuing telemetry transport) protocol as the core is designed. The system can adjust the topology to adapt to different production scales. All controllers adopt the same interface. The controller is not limited to temperature controller, and can expand various types of upper computers. In addition, the security authentication program of the Internet of things system is designed.

Finally, a web client is designed to provide login users with control panels for different types of controllers, display the state of the controller with a graphical interface, and issue control commands to the controller.

KEYWORDS: Agricultural informatization; Internet of things; Temperature control; MQTT; Web client

目录

- 1 绪论.....1
 - 1.1 研究的内容.....1
 - 1.1 研究的背景和意义.....1
 - 1.2 论文结构安排.....4
- 2 系统结构设计.....5
 - 2.1 系统整体结构.....5
 - 2.2 网络拓扑.....7
 - 2.2.1 适合小生产规模的拓扑.....7
 - 2.2.2 适合较大生产规模的拓扑.....8
 - 2.2.3 两种拓扑的适用场景.....8
 - 2.3 控制器硬件设计.....9
 - 2.3.1 整体结构.....9
 - 2.3.2 电源模块的硬件结构.....10
 - 2.3.3 驱动单元的硬件结构.....13
 - 2.3.4 ESP32模组的外围电路设计.....14
 - 2.3.5 MSP430的外围电路设计.....15
 - 2.4 本章小结.....17
- 3 水箱温度模型建模与控制算法仿真.....18
 - 3.1 水箱温度模型的机理建模.....18
 - 3.2 参数辨识.....20
 - 3.3 控制算法仿真.....20
 - 3.3.1 抗积分饱和.....21
 - 3.3.2 PI.....22
 - 3.3.3 模糊PI.....23
 - 3.3.4 带史密斯预估器的PI.....27
 - 3.3.5 带史密斯预估器的模糊PI.....28
 - 3.3.6 四种控制算法效果对比.....29
 - 3.4 本章小结.....30
- 4 通信协议设计.....31
 - 4.1 MSP430与ESP32的通信协议.....31
 - 4.2 MQTT协议简介.....33
 - 4.3 对MQTT客户端ID格式的规定.....34
 - 4.4 对MQTT主题的规划.....34
 - 4.4.1 子主题.....35
 - 4.4.2 主场、客场.....35
 - 4.4.3 主场发布.....35
 - 4.4.4 客场发布.....35
 - 4.4.5 主场接收.....36
 - 4.4.6 客场接收.....36
 - 4.5 上位机和控制器之间的通信.....36
 - 4.5.1 控制器发送消息的方式.....36
 - 4.5.2 控制器接收消息的方式.....37
 - 4.5.3 上位机发送消息的方式.....37
 - 4.5.4 上位机接收消息的方式.....38
 - 4.6 本章小结.....38
- 5 系统软件设计.....39
 - 5.1 采用的平台或框架.....39
 - 5.2 控制器的接口实现规范.....39
 - 5.3 减小MQTT主题匹配的时间复杂度.....40
 - 5.4 HTTP服务器的接口和安全.....41
 - 5.4.1 登录认证接口.....41
 - 5.4.2 MQTT协议的安全性.....42
 - 5.4.3 MQTT连接认证接口.....45
 - 5.4.4 MQTT订阅和发布认证接口.....47
 - 5.4.5 获取设备信息接口.....49

5.5 网页端用户登录认证	49
5.6 设备列表组件	51
5.7 ESP32组件	53
5.7.1 在线状态显示	53
5.7.2 温度显示	54
5.7.3 LED开关	55
5.7.4 日志显示	55
5.8 TestDevice组件	56
5.8.1 在线状态指示灯	56
5.8.2 日志显示	56
5.8.3 发送按钮	56
5.9 通信功能测试	57
5.9.1 以test-1的身份与网页进行交互	57
5.9.2 以上位机的身份与控制器进行交互	60
5.10 本章小结	61
6 总结与展望	62
6.1 总结	62
6.2 展望	62
参考文献	64
致谢	66
插图清单	
图 2.1: 系统组成框图	5
图 2.2: 控制器作为上位机	6
图 2.3: 适用与小生产规模的拓扑	7
图 2.4: 适合较大生产规模的拓扑	9
图 2.5: 控制器整体结构	10
图 2.6: 电源模块结构图	11
图 2.7: 市电经整流桥输出220V直流	11
图 2.8: 12V输出硬件原理图	12
图 2.9: 3.3V输出硬件原理图	12
图 2.10: 驱动单元硬件原理图	13
图 2.11: ESP32外围电路	14
图 2.12: 将ESP32的RXD0和TXD0通过排针引出	15
图 2.13: ESP32和MSP430的UART连接	15
图 2.14: MSP430整体外围电路	15
图 2.15: MSP430左侧引脚连接	16
图 2.16: 1-wire-bus网络引出到排针	16
图 2.17: MSP430右侧引脚连接	17
图 3.1: 用于舍入的matlab function模块	21
图 3.2: 抗积分饱和程序	21
图 3.3: PI控制算法仿真模型图	22
图 3.4: m脚本中的全局的PI控制器参数的结构体	22
图 3.5: PI控制算法仿真模型输出	23
图 3.6: 模糊PI控制算法仿真模型图	23
图 3.7: e的隶属度曲线	24
图 3.8: ec的隶属度曲线	24
图 3.9: kp的隶属度曲线	24
图 3.10: ki的隶属度曲线	25
图 3.11: 模糊PI仿真模型输出	26
图 3.12: 带史密斯预估器的PI控制仿真模型	27
图 3.13: 带史密斯预估器的PI控制器参数	28
图 3.14: 带史密斯预估器的PI控制仿真模型输出	28
图 3.15: 带史密斯预估器的模糊PI控制算法仿真模型	29
图 3.16: 四种控制算法输出 (曲线1: 带史密斯预估器的PI, 曲线2: 带史密斯预估器的模糊PI, 曲线3: PI, 曲线4: 模糊PI)	30
图 4.1: MQTT的订阅/发布模型示意图	33
图 5.1: 初始化第三个子主题的map节点	40
图 5.2: 查找map调用对应的函数	41
图 5.3: 用户表	42
图 5.4: 登录认证接口	42
图 5.5: 第一个窗口出现的提示	43

图 5.6: MQTT安全认证.....	45
图 5.7: 设备表.....	46
图 5.8: MQTT连接认证流程图.....	47
图 5.9: 订阅/发布认证流程.....	48
图 5.10: 获取设备信息接口.....	49
图 5.11: 登录界面.....	50
图 5.12: 网页客户端用户登录流程.....	50
图 5.13: 重定向.....	51
图 5.14: 设备列表界面.....	52
图 5.15: C# 和html互相嵌套的语法.....	53
图 5.16: ESP32组件界面.....	53
图 5.17: ESP32离线后更新指示灯.....	54
图 5.18: LED开关的控制逻辑.....	55
图 5.19: TestDevice组件界面.....	56
图 5.20: 设置遗嘱.....	57
图 5.21: 建立与MQTT Broker的连接.....	57
图 5.22: 订阅主题.....	58
图 5.23: 收到的消息.....	58
图 5.24: 控制器回复之前和回复之后的在线状态指示灯.....	58
图 5.25: 告诉网页自己在线.....	58
图 5.26: 发送日志信息.....	58
图 5.27: 网页中显示了日志信息.....	59
图 5.28: MQTTfx收到问候消息.....	59
图 5.29: 问候消息内容.....	59
图 5.30: 关闭MQTTfx后在线状态指示灯的变化.....	60
图 5.31: 上位机的身份连接到MQTT服务器.....	60
图 5.32: 收到控制器的温度信息.....	60
表格清单	
表 3.1: kp参数的模糊规则.....	25
表 3.2: ki参数的模糊规则.....	26
表 4.1: 串口通信帧.....	31
表 4.2: 应用层帧格式.....	32
表 4.3: 写LED状态.....	32
表 4.4: 读LED状态.....	32
表 4.5: MSP430返回给ESP32的表示LED灯状态的帧.....	32
表 4.6: MSP430发给ESP32的温度信息.....	33

绪论

研究的内容

本课题设计一个基于物联网的温度控制系统，分为两部分，一部分是温度控制器，另一部分是物联网系统。从底层的控制器到上层的服务器、客户端都进行了设计，涉及的内容有：控制器硬件设计、控制器的通信程序设计、控制算法仿真、HTTP服务器程序设计、SQL、MQTT协议、网页设计、安全认证、网络拓扑结构设计。

温度控制器以水床孵化箱为背景，针对养鸡业的种蛋孵化对温度的小超调、温度平稳、控制较精确的要求，分别对PI、模糊PI及两者与史密斯预估器结合的控制算法进行了仿真，对仿真的结果进行了对比分析。此外，进行了控制器的硬件设计。

以MQTT（消息队列遥测传输）协议为核心，设计了一个统一接口的，可扩展的物联网系统。围绕着MQTT，各种类型的控制器都能够以相同的形式实现接口，各种类型的上位机都能够通过用一种方式和控制器进行通信。系统可以添加各种功能的上位机（例如：收集各个控制器状态信息上位机、为控制器提供各种服务的上位机、不同平台的图形界面客户端等），能够添加新类型的控制器。控制器只需维护一个MQTT连接即可实现与任意个上位机同时通信。此外，系统可以调整网络拓扑结构以满足不同生产规模的用户的需求。

针对MQTT协议在安全方面的不足，设计一种认证方式，解决了两个安全问题。此外设计了HTTP服务器的接口安全认证程序。

网页客户端以Blazor为框架，使用组件化设计，以组件的形式对功能进行扩展，给用户提供了一个与控制器进行交互的平台。网页客户端中每个控制面板组件独立与控制器进行通信，互不关联。

1.2.1 研究的背景和意义

2012年11月，中共十八大提出，在中国共产党成立100年时全面建成小康社会，在新中国成立100年时建成富强民主文明和谐的社会主义现代化国家。2020年，我国已实现第一个百年目标，正在朝第二个百年目标迈进。全面建设社会主义现代化国家，实现中华民族伟大复兴，最艰巨最繁重的任务依然在农村，最广泛最深厚的基础依然在农村。[1]

2022年2月22日，农业农村部印发了《“十四五”全国农业农村信息化发展规划》（以下简称《规划》），对“十四五”时期农业农村信息化高质量发展做出系统部署。《规划》提出，智慧农业是“十四五”时期农业农村信息化发展的主攻方向，重点是聚焦行业发展需求，提升农业生产效率。从智慧种业、智慧农田、智慧种植、智慧畜牧、智慧渔业、智能农机和智慧农垦七个方面进行全面突破。[2]

目前，我国农业农村信息化发展仍处于起步阶段，还存在几个方面的短板。一是网络基础设施不足，一些偏远的农业生产区

域尚未实现网络覆盖;二是创新能力不足,关键核心技术亟待突破,先进适用的信息化产品装备缺乏;三是有效数据不足,用数据支撑生产经营和管理决策的作用不够;四是人才不足,缺乏既懂三农又懂信息技术的复合型人才。[3]

物联网技术是继计算机技术和互联网以及移动通信网络之后的又一次信息化浪潮,能够实现物与物、人与物之间的连接互动。这极大地提高了各个领域的生产生活效率,转变了传统的生产生活方式[4]。现阶段,较为普及的物联网应用场景是智能家居,在国内有:海尔、美的、格力、华为、小米、易微联等企业。这些企业的智能家居产品使用方便,有易操作的用户图形界面,安装简单,无需专业技能。但是,物联网在农业方面的应用现阶段没有普及,《“十四五”全国农业农村信息化发展规划》指出:“现阶段,农业的信息化装备缺乏,网络基础设施不足”。所以,物联网在农业方面的应用有很大的发展空间。

针对这个发展现状,本设计设计一个基于物联网的温度控制系统,控制对象为水床孵化箱。

在工业中,控制网络采用DCS或FCS,最后由上位机或网关来接入互联网。这种结构是企业设计的,面向的是有一定规模的工业现场,而个人经营的小工厂,设备数量不多,没有布置DCS或FCS的必要和能力。我国现阶段的土地政策仍是家庭联产承包责任制,个体农户居多,存在少量的农业合作社和农业企业,而个体农户生产规模较小,不适用DCS或FCS。要让物联网在农业中普及,不能抛弃小农户,所以需要一种像智能家居的网络拓扑结构那样的农业物联网。此外,在智能家居领域,各种品牌的物联网设备之间接口互相不兼容,不能相互操作,行业没有统一的标准,这为用户带来了困扰。制作农业物联网设备需要掌握物联网技术,同时需要了解农业的背景。农业涉及面较广,各种作物、家畜、家禽的种植、养殖都有相关的知识,一家企业难以做到面面俱到,而公平竞争的市场环境也会导致像智能家居那样出现多家企业,使用多种标准。此外,农户的经营并不专一,会根据不同时期种植不同作物,同时养殖多种家畜、家禽,如果像智能家居那样各种设备使用不同的接口,互相不兼容,没有统一的管理平台,会对生产形成阻碍。因此,农业物联网需要具有统一的接口和管理平台。

与工业现场相比,农业现场的通信环境良好,没有工业现场中的众多干扰源,因此抗干扰性能无需像工业控制网络那么好,这使无线通信成为可能。

基于此,本设计设计一个统一接口的,能够调整拓扑结构以适应不同生产规模的物联网系统。本系统以MQTT为核心,为低性能控制器提供可靠的通信,可降低控制器的成本。

传统的母鸡孵化受母鸡的能力影响,卫生条件难以保障,在鸡舍开放环境中容易遭受鼠害导致减产,在孵化后期,未受精种蛋会变质,如果不及时清理,容易导致雏鸡生病,严重时还会影响孵化成功率。采用孵化设备孵化便于管理,能够保障卫生,能在孵化前期检测出未受精种蛋和在中后期清理变质种蛋。种蛋对于温度要求较严格,全程温度需要保持在37.5℃到38.5℃的范围内,孵化期为21天。[5],随着时间的推移,还需逐步降低温度,温度过高容易导致水分蒸发过快,降低孵化和破壳的成功率,严重时会导致胚胎失活;温度过低会导致孵化时间延长,因此要求对温度的控制较精确。同时,因为水的比热容较大,为了节约能源,孵化箱的保温效果较好,所以如果出现超调,执行机构只有加热能力,没有散热能力,自然散热需要的时间较长,胚胎将长时间处于过温条件下。而孵化周期长达21天,所以该温度控制系统对动态性能要求不高,对稳态性能要求较高,且需要尽量抑制甚至消除超调。

由于生物对环境具有适应能力,同时个体之间有差异,所以最适合的孵化温度是不精确的,只能给出一个范围,因此,不必对控制的精度过分要求,只要能满足温度在适合的范围内,同时有微调的余地就行。

当前市面上的孵化箱大多采用基于回差温度带的开关量控制,在纯滞后较大时,难以保证控制质量,会降低孵化成功率。

本项目中对PI控制、模糊PI控制及两者与史密斯预估器的结合进行了仿真和效果的对比分析,从仿真结果来看,最终的温度较稳定,稳态误差符合要求,能够提供更好的孵化温度条件。

1.2.2 论文结构安排

第一章为绪论,介绍了该项目的研究内容、研究背景和研究的意义。

第二章中介绍了本项目的系统结构设计,首先介绍系统的整体结构,接着以此结构为基础设计了两种拓扑结构,第二部分为控制器的硬件结构和硬件选型,进行了硬件原理图的设计和硬件选型。

第三章为水箱温度模型的建模与控制算法的仿真,首先通过机理建模得出水箱温度模型,接着进行参数辨识从而进行控制算法的仿真。

第四章为软件设计,进行了HTTP服务器、ESP32、MSP430、网页等各部分的软件设计和,介绍了对MQTT协议的使用,设计了HTTP服务器接口和对MQTT客户端的行为的安全认证程序,使用MQTTfx模拟一个控制器与网页进行交互,使用MQTTfx模拟一个上位机与控制器进行交互。

第五章为总结与展望,总结了本项目所做的工作,本项目的背景和意义,并说明了本项目中的不足及能够进行改进的地方。

-----绪论-----

1.3 系统结构设计

本章中“系统的整体结构”一节介绍了系统的组成、各组成部分之间的关系和它们的功能。“网络拓扑”一节提出了适用于小规模 and 较大规模的生产环境的两种拓扑结构。“控制器的硬件结构”一节先介绍了控制器的整体硬件结构和各部分的功能,然后详细介绍各部分的结构和硬件原理图。

1.3.1 系统整体结构

本系统由HTTP服务器、MQTT服务器、云计算服务器、客户端、控制器、除了客户端以外的上位机组成。系统的组成框图如图 2.1所示。

图 2.1: 系统组成框图

HTTP服务器中含有HTTP接口、SQL、静态文件服务。静态文件服务用于让用户获取到网页,SQL用于储存用户和控制器的信息。HTTP接口用来提供登录认证、MQTT连接认证、MQTT订阅/发布认证、获取控制器信息的服务。

客户端是一个HTTP客户端。一方面,如果客户端是网页客户端,需要向HTTP服务器获取网页的静态文件,这时用到了静态文件服务;另一方面,客户端在用户登录时需要用到HTTP服务器提供的“登陆认证接口”,在用户登录成功后,需要向用户呈现其所拥有的控制器的控制面板,这需要先使用HTTP服务器提供的“获取设备信息接口”来获取用户的设备。

所有的上位机和控制器都是MQTT客户端,它们之间的通信都依靠MQTT 服务器进行转发。上位机包括客户端和其他不是客户端的上位机。不是客户端的上位机种类可以有很多,例如:云计算服务器、控制器等。甚至MQTT通信的调试工具也可以成为一

个上位机。

控制器也可以是其他一些控制器的上位机。以孵化车间为例，底层的控制器是温度控制器，用于实时控制孵化箱的温度，还有一个控制器作为它们的上位机，内部有实时时钟系统，用于在不同的孵化天数向底层的温度控制器发送命令，微调给定温度。这时客户端、作为上位机的控制器、底层的控制器之间的关系如图 2.2所示

图 2.2: 控制器作为上位机

云计算服务器可以提供控制算法服务，让控制器对控制算法进行计算卸载。控制器采用的是低级的计算机，计算能力和存储空间有限，无法运行高级的控制算法，利用云计算服务器可以提高控制质量。此外，控制算法的输入和输出信息量少，输入信息为给定值和反馈信号，输出信息为执行机构的输出值，而过程控制对实时性要求不高，很适合使用计算卸载。在网络出现故障时，可以用控制器内部的低级的控制算法进行取代控制。

无论有多少个上位机，控制器同时与所有的上位机进行通信都只需要维护一个MQTT连接，与对每个上位机维护一个TCP连接相比，大大减小了对控制器的压力。

对控制器来说，与所有类型的上位机进行通信没有区别，都是通过MQTT的订阅/发布的模式，这使得控制器无需对每一种类型的上位机实现一套接口，大大简化了程序的设计。同时，这也使得上位机无论面对什么控制器，只要该控制器是基于MQTT协议的，都能使用相同的方式与之通信，各种类型的控制器的接口统一，简化了上位机程序的开发。

因为MQTT没有对客户端ID和主题的模式做出限制，自由度很大，为了让上位机和控制器有序通信，便于管理，控制器能够有统一的接口，上位机能够以相同的方式利用这些接口，需要对MQTT的客户端ID和主题做出一些格式上的规范。对客户端ID的规范见4.3节，对主题的规范见4.4节。

MQTT服务器最核心功能为MQTT Broker (MQTT代理)，MQTT客户端之间不能直接通信，所有消息帧都需要经过MQTT代理的转发。但是，MQTT服务器的功能不局限于MQTT代理。以MQTT服务器软件emqx为例，emqx是一个开源的MQTT服务器软件，它除了提供MQTT Broker服务以外还集成了一些插件。例如本项目用到的HTTP认证插件。使用该插件后，emqx也将作为一个HTTP客户端，当需要对MQTT客户端的行为进行认证时，emqx将使用HTTP服务器提供的接口。

1.3.2 网络拓扑

依靠MQTT协议，并对客户端ID、主题做出规范后，本系统具有很大的灵活性和可扩展性。可扩展性表现在可以方便地加入各种类型的上位机，各种类型的控制器。灵活性表现在可以方便地调整拓扑结构以适应不同生产规模的需求。

1.3.2.1 适合小生产规模的拓扑

图 2.3: 适用与小生产规模的拓扑

个体农户的生产规模较小，非专业的，不以盈利为目的的私人圈养的家禽也是小生产规模用户的重要组成部分。这部分用户的特点是每个人拥有的控制器数量较少，但是人数众多。这种场景和智能家居类似，一般家庭不会布置自己的智能家居服务器。智能家居采用的是云端的公共的服务器，所有用户都使用该服务器，通过手机APP来管理自己的设备，所有的智能家居设备都通过互联网连接到服务器。在小规模生产时，本系统也可以调整为与智能家居类似的拓扑结构。如图 2.3所示。

在这种拓扑下，系统的所有组成部分都连接到互联网。HTTP服务器、MQTT服务器和以云计算服务器为代表的其他类型的上位机为来自各地的，互联网上的用户和他们的控制器提供服务。服务器由专业的企业维护，以出租服务向用户收费的方式盈利。这种拓扑的缺点是不可定制，只能使用服务提供商提供的服务和硬件。

1.3.2.2 适合较大生产规模的拓扑

对于生产规模较大的农业合作社或企业，这些单位有自己的物联网和控制算法方面的专业团队，希望能够定制自己的系统，例如希望添加一些除了客户端以外的其他类型的上位机，希望自己制作网页，希望接入自己设计的控制器，对公共的服务器安全性不放心，对于公共服务器的租赁价格不满意等，这时就可以使用这种拓扑结构。如图 2.4所示。

在这种拓扑中，除了网页客户端在互联网上，系统的其余组成部分都位于局域网内。

1.3.2.3 两种拓扑的适用场景

小生产规模的用户没有维护服务器的能力，为少量设备单独建立服务器的性价比不高。所以小生产规模的用户适用“适合小生产规模的拓扑”。大生产规模的用户，如果没有高度定制系统的需求，又没有对公共服务器的安全性的担忧，对租赁价格也满意，也可以使用“适合小生产规模的拓扑”。但是，“适合较大生产规模的拓扑”不推荐小生产规模的用户使用。

图 2.4: 适合较大生产规模的拓扑

1.3.3 控制器硬件设计

本设计要研究的控制器是水床孵化箱中的水箱温度控制器。需要采集水箱温度，进行控制算法计算后输出控制量来驱动执行机构，此外，还要通过MQTT协议接收来自上位机的控制命令和将状态信息发送给网页。

下面将介绍控制器的硬件结构。首先介绍整体结构，然后分模块进行详细介绍。

1.3.3.1 整体结构

控制器的作用是检测水箱温度，进行控制算法的计算，输出控制量来控制水箱温度。此外还要与网页客户端通信。控制器整体由电源模块、驱动单元、MSP430、ESP32、铠装DS18B20探头组成。如图 2.5所示。

电源模块接收220VAC市电的输入，提供3路电压输出，分别是12VDC、220VDC、3.3VDC。220V和12V电压都是用来供给驱动单元。3.3V电压用来供给MSP430、ESP32和铠装DS18B20探头。

驱动单元有光耦隔离驱动模块、IGBT模块，IGBT的集电极电流用来驱动外接的电热管进行加热。

DS18B20是数字式的温度传感器，铠装使得传感器能够投入水中进行温度采样。

MSP430负责温度采样、控制算法的计算和控制量输出。ESP32的功能是使用MQTT协议与网页进行通信，使用UART与MSP430进行通信，作为网关对MQTT协议和建立在UART之上的应用层协议进行转换。

图 2.5: 控制器整体结构

1.3.3.2 电源模块的硬件结构

电源模块的结构图如图 2.6所示。电源模块的输入为220VAC市电，通过进线端子引入后分为两路，一路输入到整流桥，一路输入到输出为5V的AC/DC隔离型直流稳压电源。

电源模块有3路输出，一路为220VDC，一路为12VDC，一路为3.3VDC。其中220VDC是50Hz的220VAC经全波不控整流得到的结

果，220VDC的输出不接电容器。12VDC是AC/DC电源的稳压输出，该输出与交流侧没有进行隔离，12VDC电源的输入为整流桥输出的220VDC。3.3VDC输出是220VAC经过输出电压为5V的AC/DC隔离型稳压电源后得到5VDC，再经过AMS1117得到的。该路输出与交流侧进行了隔离。

3. 3V直流电用于给控制器的芯片供电，12V和220V直流电用于给控制器的驱动单元供电。

图 2.6：电源模块结构图

220V直流输出的硬件原理图如图 2.7所示

图 2.7：市电经整流桥输出220V直流

整流桥选用GBJ2004-F，该整流桥的反向重复峰值电压为400V，在60Hz时的额定平均电流为20A。使用市电输入，输出电流达到最大的20A时可得到4400W的输出功率。正向浪涌电流可达240A。本项目中220V直流输出用于经过IGBT的斩波后驱动电热管，输出端不使用电容器与电热管并联，除非操作失误导致短路，否则不会出现浪涌电流。

12V直流输出的硬件原理图如图 2.8所示。

图 2.8：12V输出硬件原理图

整流桥的220V直流输出输入金升阳LS03-K3B12SS后输出12V直流电。该电源为稳压输出，额定输出功率为3W，额定输出电压为12V，可接受70V到430V直流输入。该电源为非隔离型电源，12V直流输出端与整流桥的220V输出的负端共地。输出纹波最大为150mV，典型值为80mV。该路输出用于给光耦供电，光耦用于驱动IGBT和起隔离作用。

3. 3V输出的硬件原理图如图 2.9所示

图 2.9：3.3V输出硬件原理图

市电经过TAS5-5-WEDT后输出5V直流电。该输出为稳压输出，输出端与交流侧隔离。

指 标		
疑似剽窃文字表述		
<div>1. 中国共产党成立100年时全面建成小康社会，在新中国成立100年时建成富强民主文明和谐的社会主义现代化国家。2020年，</div> <div>2. 《规划》）提出，智慧农业是“十四五”时期农业农村信息化发展的主攻方向，重点是聚焦行业发展需求，提升农业生产效率。</div>		
2. 基于物联网的温度控制系统设计_第2部分		总字数：9946
相似文献列表		
去除本人文献复制比：0.8%(75) 文字复制比：0.8%(75) 疑似剽窃观点：(0)		
1	基于COBS协议的自适应最优帧长分析 李江(导师：孙振东;左巍) - 《华南理工大学硕士论文》- 2013-05-01	0.7% (72) 是否引证：否
2	基于COBS协议的自适应最优帧长分析 - 道客巴巴 - 《互联网文档资源 (http://www.doc88.com) 》- 2019	0.7% (72) 是否引证：否
原文内容		

TAS5-5-WEDT的额定输出电流为1A，额定输出输出功率为5W。输入端可接受直流输入和交流输入，交流输入的范围为85V到265V，隔离电压达2500V交流。输出纹波噪声为50mV。TAS5-5-WEDT的输出作为AMS1117的输入，最后输出3.3V的直流电。该输出用于给MSP430，ESP32，DS18B20供电。该款AMS1117的额定输出电流达到1A，额定输出功率为3.3W。MSP430G2553在引脚不输出电流的条件下消耗的的最大电流为420μA，在本项目中，MSP430没有引脚使用强驱动输出。ESP32模组的用户手册建议的电流最小值为0.5A，DS18B20的工作电流为1.5mA。综上使用AMS1117CD-3.3足够供电。

1.3.3.3 驱动单元的硬件结构

驱动执行单元接受MSP430输出的PWM信号，最终输出受调制的220V直流电，用于驱动电热管。驱动单元的硬件原理图如图2.10所示

图 2.10：驱动单元硬件原理图

TLP250H(F)为光耦。该光耦既有隔离的作用，也有驱动的作用。该光耦的输出侧采用12V直流稳压电源供电。输出侧采用的是两个MOSFET组成的半桥结构。当输入侧的发光二极管没点亮时，下半桥导通，上半桥截止，光耦输出端输出低电平，当输入侧的发光二极管点亮时，上半桥导通，下半桥截止，光耦输出端输出高电平。

该光耦的高电平最小值比供电电源的VCC低4V，低电平的最大值比供电电源的VEE高2.5V，在VCC=12V，VEE=0V时，输出高电平最小值为9.5V，输出低电平最大值为2.5V。

使用的IGBT为STGF19NC60KD。该款IGBT在VCE = VGE时，由截止状态到导通状态的栅极-射极门槛电压VGE(th)最大值为6.5V，使用光耦TLP250H(F)在12V直流供电时能够驱动该款IGBT。

光耦的输入端的发光二极管的阳极接MSP430的PWM输出，阴极通过电阻接地。该地为3.3V直流输出的地。IGBT的射极接到整流桥输出的220V直流的负端，这也是12V直流输出的地。通过让光耦的工作电源和IGBT的射极共地，使得无需使用自举电路即可驱动IGBT。整流桥输出的220V直流的正端接到电热管端子的一端。电热管接线端子的另一端连接IGBT的集电极。当电热管的两条线接入端子后，电流从整流桥输出的220V的正端流经电热管后进入IGBT的集电极，流出IGBT的射极后回到220V直流的负端。

通过将IGBT串联进该回路，使得电热管两端的电压受到IGBT的调制，从而驱动电热管并控制电热管的加热功率。

TLP250H(F) 光耦的供电电压可为10V到30V。没有输出电流时，供电电流最大为3mA，输出电流最大为 $\pm 2.5A$ 。IGBT的栅极电流为 $\pm 100nA$ ，使用金升阳LS03-K3B12SS能够给该光耦供电，光耦的输出电流也足够驱动IGBT。

该款IGBT的TO-220FP封装类型在100℃时的集电极最大电流为10A，栅极电流为 $\pm 100nA$ ，集电极射极可承受的最大电压为600V。在散热条件良好，能够使该IGBT温度维持在100摄氏度以下的情况下使用该IGBT驱动电热管，最高可获得2200W的功率。

1.3.3.4 ESP32模组的外围电路设计

ESP32的外围电路如图 2.11所示。

图 2.11: ESP32外围电路

ESP32的外围电路中有复位按钮和BOOT按钮。这两个按钮可以使ESP32进入下载模式。IO0为BOOT引脚。在ESP32被复位后的一段时间内，如果IO0引脚被拉低，或者先将IO0引脚拉低，再复位ESP32，就会使ESP32进入BOOT模式。在该模式下，可以通过UART将程序下载到ESP32模组内部集成的flash模块中。下载完成后再次复位，ESP32将从flash启动，将flash的代码装载到RAM中开始运行。ESP-TX0和ESPRXD0为下载和调试引脚，通过排针引出。如图 2.12所示

图 2.12: 将ESP32的RXD0和TXD0通过排针引出

ESP-RXD1和ESP-TXD1为ESP32模组的第二个UART的输入和输出引脚。这两个引脚用来与MSP430进行通信。ESP-TXD1和MSP-RXD连接，ESP-RXD1和MSP-TXD连接。如图 2.13所示

图 2.13: ESP32和MSP430的UART连接

ESP32的3.3V电源和地来自电源模块中的3.3V直流输出。

1.3.3.5 MSP430的外围电路设计

MSP430选用双列直插的MSP430G2553IPW28。整体外围电路如图 2.14所示。

图 2.14: MSP430整体外围电路

左侧的引脚连接如图 2.15所示。右侧的引脚连接如图 2.17所示。

图 2.15: MSP430左侧引脚连接

左侧引脚的DVCC连接到电源模块的3.3V输出，MSP-RXD和MSP-TXD用来和ESP32进行通信，分别和ESP-TXD1，ESP-RXD1进行连接，连接图如图 2.13所示。1-wire-bus是一线总线，一线总线协议是DS18B20的通信协议。该协议利用一条信号线实现主从的半双工通信。DS18B20始终作为从机。使用该协议需要严格遵守时序。该引脚需要通过 $5k\Omega$ 电阻上拉到3.3V。DS18B20在使用独立的电源线供电时信号线不需要强上拉，信号线只需连接MSP430的一个引脚。该网络连接到排针的引脚中，如图 2.16所示。

图 2.16: 1-wire-bus网络引出到排针

在排针处，该网络已完成上拉。同时，3.3V的电源输出和地也已经引出到排针。图 2.16的1，2，3引脚是为DS18B20准备的。引脚1连接到DS18B20的信号线，引脚2连接到DS18B20的VCC引出线，引脚3连接到DS18B20的GND引出线。最终，将铠装DS18B20探头的3条先通过手工焊接的方式连接到这3个引脚。

在图 2.16中还将MSP-TXD和MSP-RXD通过排针引出。这是为了下载程序。MSP430下载程序通过ezFET接口，此接口还有两个引脚，分别为MSP-RST# 和MSP-TEST，这两个引脚也通过排针引出，如图 2.12所示。

PWM-OUT网络的另一处连接点是驱动单元的光耦的输入侧的发光二极管的阳极，如图 2.10所示。该网络连接到MSP430的定时器的PWM输出引脚，用来输出PWM信号来驱动光耦的发光二极管，通过光耦驱动IGBT，通过IGBT对电热管两端的电压进行调制，以达到调节电热管加热功率的目的。

MSP430右侧的引脚连接如图 2.17所示。右侧的XIN和XOUT分别连接着32768Hz的石英晶体振荡器的两端。MSP430内部有集成的，可以通过软件选择大小的电容器，无需外接电容。右侧有复位按钮。ezFET接口有自动下载和复位的功能，无需手动通过按钮使MSP430进入下载模式。

图 2.17: MSP430右侧引脚连接

MSP430的TEST引脚连接到MSP-TEST网络，RST# 引脚连接到MSP-RST网络。这两个引脚是ezFET接口的引脚，RST# 是复位引脚。在上文中已讲述了将这两个引脚通过排针引出以方便下载程序和调试。

1.3.4 本章小结

本章介绍了本设计的系统的整体结构，以此为基础，介绍了两种拓扑结构及其适用的场景，接着介绍了控制器的硬件设计和硬件选型。

系统结构

1.4 水箱温度模型建模与控制算法仿真

本章使用机理建模的方法建立水箱温度模型，并进行了参数辨识，最后对4种控制算法进行了仿真和对比分析。

1.4.1 水箱温度模型的机理建模

水的比热容为 c ，质量为 m ，热量与温度的关系为 $dQ=cm\cdot dT$ ， Q 为热量， T 为温度。如果不考虑散热，只考虑加热，则 Q 与加热功率的关系如公式(4.1)所示

(4.1)

由方程组(4.2)(4.2)

得公式(4.3)(4.3)

对公式(4.3)两边进行拉氏变换，得公式(4.4)(4.4)

则此时的水箱温度模型是一个积分环节。

如果考虑散热，水箱的散热是一个非常复杂的模型。水箱中的水的散热方式有：热传导、蒸发、热辐射。水箱上方的气体的对流情况也会影响散热速度。此外环境温度、空气相对湿度、大气压也会影响水箱的散热。但是，可以肯定的是其他条件不变时，水的温度越高，散热越快，在平衡点附近取泰勒级数的一次项对水箱散热功率进行线性化，使整个系统线性化。

考虑散热，并且将水箱视为集总参数模型，则系统的方程如公式(4.5)所示(4.5)

解方程组(4.5)，得公式(4.6)(4.6)

设平衡时，温度为，加热功率和散热功率分别为，，则平衡时，系统的方程如公式(4.7)所示(4.7)

在平衡点的基础上，将系统的方程表示为偏差量，得到公式(4.8) (4.8)

由公式(4.7)和公式(4.8)得公式(4.9) (4.9)

接下来要对散热功率进行线性化。设散热功率和温度的关系为：，在处展开为泰勒级数后表示为公式(4.10) (4.10)

只保留一次项，得到公式(4.11) (4.11)

将表示为，表示为 ΔT ，设则(4.12)

由公式(4.9)和公式(4.12)得公式(4.13) (4.13)

对公式(4.13)两边进行拉氏变换得公式(4.14) (4.14)

化简公式(4.14)得公式(4.15) (4.15)

其中 $K=1/a$ ， $T=cm/a$ ， a 的量纲为 s 是秒， J 是焦耳。

现在得到了水箱得温度模型，是一个一节惯性环节。考虑到水的热传导和对流需要时间，需要再加上一个纯滞后环节。则水箱温度模型为公式(4.16) (4.16)

1.4.2 参数辨识

一个16W的加热杯垫，使用不锈钢杯。环境温度为25.25℃，将水静置，等到温度稳定后，最终的温度为55.13℃。计算出温升为29.88℃。由公式(4.12)解得 $a=0.49$ 。水的体积取4L，换算为质量为4kg，根据公式(4.15)，求得 $K=2.04$ ，纯滞后时间 τ 取得尽量夸张，取300秒。实际应用中纯滞后时间如果大于300秒，可以在水箱内布置热管，在加快热传递速度的同时使得水箱内部温度分布更加均匀。

最终得到的得到水箱的温度模型，如公式(4.17)所示(4.17)

1.4.3 控制算法仿真

本节对4中控制算法进行仿真，分别是：“PI”，“模糊PI”，“带史密斯预估器的PI”，“带史密斯预估器的模糊PI”。最后对4种算法进行了对比。四种算法中的PI控制器部分都通过S函数实现，采样周期为10s。由于本系统选用的IGBT的最大功率为2200W，所以仿真时将执行机构最大输出功率定为2000W。

图 3.1：用于舍入的matlab function模块

由于DS18B20的精度为1/16℃，为了接近真实环境，使用matlab function模块在反馈通道上对反馈值进行舍入，将精度保留为1/16℃。该matlab function模块如图 3.1所示。为了模拟真实环境中执行机构输出值有上下限的情况，在控制器的输出端使用限幅模块进行限幅。在实现PI控制的s函数中，还进行了抗积分饱和的处理。

1.4.3.1 抗积分饱和

因为执行机构的输出是有范围的，当积分环节输出的值超过执行机构能够输出的最大值时就没有意义，而且还会徒增积分退保和的时间，使得超调时间变长。抗积分饱和的程序流程图如图 3.2所示。

图 3.2：抗积分饱和程序

1.4.3.2 PI

使用matlab和simulink进行仿真。PI控制算法的仿真模型图如图 3.3所示。在比较点后使用一个零阶保持器，采样时间设为10s，同时，s函数的采样时间也为10s。图中的sf3模块即为实现PI控制的s函数模块。反馈通道中的matlab function即为图 3.1所示的用于将反馈值舍入到1/16℃的模块。

图 3.3：PI控制算法仿真模型图

通过全局变量来实现工作区与s函数模块内部的函数之间的通信。在m脚本中通过结构体设置PI控制器的PI参数。通过结构体中的变量来保存输出值和积分环节的累加值。如图 3.4所示。

图 3.4：m脚本中的全局的PI控制器参数的结构体

在s函数中通过图 3.4所示的全局的结构体取得PI控制器的参数。计算完积分环节的累加值后储存到结构体的ui字段中。P字段为比例系数，I字段为积分系数。output字段为输出值。

执行仿真后，将仿真模型的输出保存到工作区，用plot函数打印图像。PI控制算法仿真模型的输出如图 3.5所示。

由于P和I足够小，该过程没有出现超调。最后达到稳态时有稳态误差，稳态误差为0.34℃。该过程的调节时间较长，长达5670s。但是，在绪论中已经讲述了，本项目针对的是水床孵化箱，孵化的时间长达21天，5670s对于21天微不足道。最后稳态时稳态误差也较小，在1℃以内，孵化温度是一个宽度为1℃的区间，使用PI控制能够满足要求。

图 3.5：PI控制算法仿真模型输出

1.4.3.3 模糊PI

模糊PI控制算法的仿真模型图如所示。

图 3.6：模糊PI控制算法仿真模型图

模糊PI的仿真模型相比PI的仿真模型，在s函数前多了一个模糊控制器，模糊控制器输出PI两个参数给PI控制器，此外，PI控制器还接受比较环节经过0阶保持器后的输出。模糊控制器的输入有两个，分别为偏差量 e 和偏差量的微分 ec 。

e 的论域为 $[-4, 4]$ ℃，偏差量的变化率的论域为 $[-2.4, 2.4]$ ℃/s。偏差量和偏差量的变化率分为7个等级，分别是PB, PM, PS, ZO, NS, NM, NB，分别代表正大，正中，正小，零，负小，负中，负大。使用matlab的模糊工具箱，生成7条均匀分布的隶属度曲线。 e 的隶属度曲线如图 3.7所示。 ec 的隶属度曲线如图 3.8所示。

图 3.7： e 的隶属度曲线

图 3.8： ec 的隶属度曲线

最左侧的隶属度曲线使用z形隶属度曲线，最右侧的隶属度曲线采用s形隶属度曲线。其他的隶属度曲线采用三角形隶属度曲线。

模糊控制器输出的 kp 的论域为 $[0, 15]$ ，使用三角形隶属度曲线。 ki 被分为5个等级，分别是：mf1, mf2, mf3, mf4, mf5。同样是利用matlab的fuzzy工具箱，生成5条均匀分布的三角形隶属度曲线。如图 3.9所示。

图 3.9： kp 的隶属度曲线

模糊控制器输出的 ki 的论域为 $[0, 0.06]$ ，也被分为mf1, mf2, mf3, mf4, mf5五个等级。使用三角形隶属度曲线，利用fuzzy工具箱生成5条均匀分布的隶属度曲线。如图 3.10所示。

图 3.10: ki的隶属度曲线

模糊推理遵循的规则为：在e较大时，kp和ki可以较大，以快速消除偏差。e较小时，ec如果太大，就要减小kp和ki，以防止超调。在e接近0时，如果ec为正且较大，就要将控制作用减到最小以防止超调。等到e接近0，且ec接近0时，增大kp和ki以减小稳态误差。当e为负时，说明正向超调了，应增大kp，出现超调很可能是因为积分环节累加得过大，应该将ki降到最小，以让超调迅速恢复。

kp的模糊规则如表 3.1所示。ki的模糊规则如表 3.2所示。

表 3.1: kp参数的模糊规则

ec	e						
	PB	PM	PS	ZO	NS	NM	NB
PB	mf5	mf5	mf5	mf5	mf5	mf5	mf5
PM	mf5	mf5	mf5	mf5	mf5	mf5	mf5
PS	mf5	mf5	mf5	mf5	mf5	mf5	mf5
ZO	mf5	mf5	mf3	mf5	mf5	mf5	mf5
NS	mf5	mf4	mf2	mf2	mf3	mf5	mf5
NM	mf5	mf4	mf2	mf1	mf3	mf4	mf5
NB	mf5	mf3	mf1	mf1	mf1	mf4	mf5

ec e
PB PM PS ZO NS NM NB
PB mf5 mf5 mf5 mf5 mf5 mf5 mf5
PM mf5 mf5 mf5 mf5 mf5 mf5 mf5
PS mf5 mf5 mf5 mf5 mf5 mf5 mf5
ZO mf5 mf5 mf3 mf5 mf5 mf5 mf5
NS mf5 mf4 mf2 mf2 mf3 mf5 mf5
NM mf5 mf4 mf2 mf1 mf3 mf4 mf5
NB mf5 mf3 mf1 mf1 mf1 mf4 mf5

表 3.2: ki参数的模糊规则

ec	e						
	PB	PM	PS	ZO	NS	NM	NB
PB	mf5	mf5	mf3	mf5	mf1	mf1	mf1
PM	mf5	mf5	mf3	mf4	mf1	mf1	mf1
PS	mf5	mf5	mf3	mf3	mf1	mf1	mf1
ZO	mf5	mf5	mf2	mf2	mf3	mf1	mf1
NS	mf5	mf4	mf2	mf1	mf2	mf1	mf1
NM	mf5	mf4	mf1	mf1	mf1	mf1	mf1
NB	mf5	mf4	mf1	mf1	mf1	mf1	mf1

ec e
PB PM PS ZO NS NM NB
PB mf5 mf5 mf3 mf5 mf1 mf1 mf1
PM mf5 mf5 mf3 mf4 mf1 mf1 mf1
PS mf5 mf5 mf3 mf3 mf1 mf1 mf1
ZO mf5 mf5 mf2 mf2 mf3 mf1 mf1
NS mf5 mf4 mf2 mf1 mf2 mf1 mf1
NM mf5 mf4 mf1 mf1 mf1 mf1 mf1
NB mf5 mf4 mf1 mf1 mf1 mf1 mf1

模糊PI仿真模型的输出如图 3.11所示。模糊PI的调节时间相比PI较短，调节时间为3530s。达到稳态后，稳态误差为0.34℃，和PI的0.35℃接近。同样能满足水床孵化箱对温度的要求。该过程也没有出现超调。

图 3.11: 模糊PI仿真模型输出

使用模糊PI后，P和I的最大值大于PI的P和I，在即将发生超调时又能够将P和I减小以抑制超调。这使得模糊PI算法的调节时间较短。

1.4.3.4 带史密斯预估器的PI

在PI控制算法和模糊PI控制算法仿真中，因为被控对象的纯滞后时间较长，惯性较大，所以不能采用较大的P和I，这导致调节时间过长。下面，将在PI控制的基础上添加史密斯预估器。

史密斯预估器是在对象模型较为准确的情况下使用的预测算法，它能够预测纯滞后时间后系统的输出值。加入史密斯预估器的PI控制仿真模型如图 3.12所示。

图 3.12: 带史密斯预估器的PI控制仿真模型

控制器的输出有两路，一路输入到被控对象，另一路输入到史密斯预估器。史密斯预估器在对被控对象进行较准确的建模后，将对象分离为两部分，一部分为不带纯滞后的系统在这里为一节惯性环节，另一部分为纯滞后环节。被控对象也是由一节惯性环节和纯滞后组成的，但是无法将被控对象的一节惯性环节和纯滞后分离开，所以采用史密斯预估器。一节惯性环节是一个线性系统，在控制器的输出同时作用于被控对象的一节惯性环节和史密斯预估器的一节惯性环节时，在相同的时间内这两个一节惯性环节的增量是相等的，无论它们的初始状态是什么。这也是史密斯预估器的工作原理。将史密斯预估器的一节惯性环节的输出减去一节惯性环节经过纯滞后环节后的输出，得到的值的物理意义就是史密斯预估器的系统输出在纯滞后时间到达后的增量，即经过纯滞后时间后，当前的史密斯预估器的纯滞后环节的输出值会在现在的基础上加上这个增量。如果对被控对象

建模较准确，这个增量也就接近被控对象在纯滞后时间之后的增量。所以，将这个值加上被控对象当前的输出，就可以得到被控对象在纯滞后时间之后的输出值。这就达到了预测的目的。

在消除了纯滞后之后，被控对象的控制就更容易了。此时，P和I可以增大。使用了史密斯预估器后的PI控制器的参数如所示。

图 3.13: 带史密斯预估器的PI控制器参数
最后，带史密斯预估器的PI控制仿真模型的输出如图 3.14所示。

图 3.14: 带史密斯预估器的PI控制仿真模型输出

带史密斯预估器的PI控制算法的调节时间达到了660s。相比PI和模糊PI，调节时间大幅减小。稳态误差为0.04℃。稳态误差也大幅减小。原因为加上了史密斯预估器使得能够使用较大的P，同时能够使用较大的I，使得消除稳态误差的速度较快。不带史密斯预估器，为了防止出现超调，只能使用极小的P和I，使得控制缓慢，同时稳态误差较大。

1.4.3.5 带史密斯预估器的模糊PI

带史密斯预估器的模糊PI的仿真模型如图 3.15所示。

图 3.15: 带史密斯预估器的模糊PI控制算法仿真模型

由3.3.4 节可知，使用了史密斯预估器后，PI控制器的P和I可以取较大的值而不用担心超调。所以，带史密斯预估器的模糊PI的模糊控制器输出的kp和ki的论域可以较广。kp的论域为 [0, 400]，ki的论域为 [0, 0.1]。

e，ec，kp，ki的隶属度曲线和3.3.3 节模糊PI相同，只有kp，ki的论域进行了调整，kp和ki的隶属度曲线仍然使用fuzzy工具箱生成均匀分布的5条三角形隶属度曲线。

带史密斯预估器的模糊PI使用的模糊规则与模糊PI使用的模糊规则相同，如表 3.1和表 3.2所示。

1.4.3.6 四种控制算法效果对比

四种控制算法的输出如图 3.16所示。曲线1为带史密斯预估器的PI，曲线2为带史密斯预估器的模糊PI，曲线3为PI，曲线4为模糊PI。

曲线1和曲线2的控制算法使用了史密斯预估器，曲线3和曲线4没有使用史密斯预估器，可以看出，使用了史密斯预估器后调节时间大幅缩短。在曲线1和曲线2达到稳态后，曲线3和曲线4远远没有达到稳态。

曲线2和曲线4使用了模糊PI，曲线1和曲线3使用了PI。可以看出，曲线2相比曲线1，调节时间有小幅缩短，但是不明显。曲线4相比曲线3，上升速度较快，最终调节时间也有缩短。根据3.3.2 节和3.3.3 节，曲线4的调节时间为3530s，曲线3的调节时间为5670s，缩短了2140s。根据3.3.4 节，带史密斯预估器的PI的调节时间为660s，使用史密斯预估器相比使用模糊控制器带来的提升更明显。而曲线4和曲线3对比起来，使用模糊控制器已经得不到太大的性能改进了。

图 3.16: 四种控制算法输出（曲线1：带史密斯预估器的PI，曲线2：带史密斯预估器的模糊PI，曲线3：PI，曲线4：模糊PI）

相比计算复杂的模糊控制器，史密斯预估器的计算较为简单，带来的提升较大。在使用性能低下的控制器时，使用带史密斯预估器的PI控制器是更好的选择。但是，史密斯预估器能够取得好的效果是建立在对被控对象的建模较准确的前提下，如果不能获得较准确的被控对象模型，史密斯预估器反而会起反作用。这时应该使用模糊PI或PI。另外，水床孵化箱对于动态性能要求不高，使用复杂的模糊控制器不一定是较好的选择，从仿真结果来看，单纯地使用PI也能够满足需求。

综上，在能够对对象建立较准确的模型时应该使用带史密斯预估器的PI控制器，反之，应该使用PI控制器，使用模糊控制器性价比较低。

1.4.4 本章小结

本章先对水箱温度模型进行机理建模，然后通过参数辨识得到实际的模型，以此模型进行了控制算法的仿真，最后分析对比了4种控制算法的控制效果，并对它们的性价比进行了分析。

-----水箱温度模型建模与控制算法仿真-----

1.5 通信协议设计

本章中介绍了ESP32和MSP430之间使用的通信协议，对MQTT协议的使用方式的规划。

通过本章中对MQTT使用的规划，使得不同的控制器能够实现相同形式的接口，控制器的一套接口可以供不同种类的上位机使用，上位机能够用相同的方式与控制器进行通信。

1.5.1 MSP430与ESP32的通信协议

MSP430与ESP32采用UART模块进行通信。在数据交互时，为了保证数据的可靠交互，需要将数据以一定的帧格式进行打包[6]。打包成帧后，帧的控制字符不能与数据中的字符混淆。为了解决这个问题，目前的协议采用的有比特填充和字节填充两种方式。HDLC协议采用的是比特填充，PPP协议采用的是字节填充。

本项目采用的是自定义协议。该协议只有帧同步的功能。该协议使用两个连续的10进制数85作为帧起始，后面紧跟着一个字节的帧长度，帧长度后面是数据域。如表 4.1所示。

表 4.1: 串口通信帧

说明	内容	字节数
帧起始	85	1
	85	1
帧长度	帧长度	1
数据域	数据	1到255

说明内容字节数

帧起始 85 1

85 1

帧长度帧长度 1

数据域数据 1到255

因为帧长度为85时会导致与帧头的第二个85组合起来被误认为是帧头，所以当帧长度为85时，需要进行转义。

相似文献列表

去除本人文献复制比：2.2%(212)文字复制比：2.2%(212)疑似剽窃观点：(0)		
1	一种基于物联网和云平台的智慧二次供水管理的实践 王菊玲;胡佳佳;肖燕;-《物联网技术》-2021-05-20	1.1%(111) 是否引证：否
2	车联网通信调度与控制教学辅助系统的设计 王涛;曾海燕;杨敏;周新国;-《滁州学院学报》-2022-04-15	1.1%(107) 是否引证：否
3	从应用场景来探讨物联网时代应急广播体系技术架构搭建 周海荣;-《中国传媒科技》-2021-06-15	1.0%(95) 是否引证：否
4	一种用于起重机械的离散数据采集分析系统系统设计 常中龙;赵丽媛;鄢鹏程;-《起重运输机械》-2021-09-15	1.0%(95) 是否引证：否
5	刘宁杰_基于安卓平台的无线蓝牙智能小车控制系统的设计与实现 刘宁杰-《大学生论文联合比对库》-2016-05-06	0.9%(90) 是否引证：否
6	6_王颖_电化学电极的测量装置设计 王颖-《大学生论文联合比对库》-2018-05-25	0.9%(90) 是否引证：否
7	基于NB-IoT温室环境远程监测系统的设计 尹晶晶;-《西安航空学院学报》-2021-09-20	0.7%(68) 是否引证：否

原文内容

在本协议中，帧长度为85时则在帧长度这个字节中填入0，接收方检测到帧长度为0就认为帧长度为85。在帧长度后面可以跟着1字节到255字节的数据。该协议有个缺点，不能传输0字节的数据。但是，本项目中的应用层不会发送0字节的数据，因为功能码域一定占一个字节，所以没有影响。

应用层的帧格式如表 4.2所示。应用层的帧被分为功能码域和数据域。

表 4.2：应用层帧格式

域	内容	字节数
功能码域	功能码	1
数据域	数据	0到254

域内容字节数

功能码域功能码 1

数据域数据 0到254

因为功能码域占用了1字节，而数据链路层的协议最多只支持发送数据域长度为255字节的帧，所以数据域最多支持254字节。

例如设置LED灯的状态，应用层的帧格式如表 4.3所示。

表 4.3：写LED状态

域	内容	含义	字节数
功能码域	1	操作LED灯	1
数据域	非零值	设置LED灯	1
	0或非零值	0表示关闭LED灯，非0值表示打开LED灯	1

域内容含义字节数

功能码域 1 操作LED灯 1

数据域非零值设置LED灯 1

0或非零值 0表示关闭LED灯，

非0值表示打开LED灯 1

读LED灯的状态，应用层的帧格式如表 4.4所示。

表 4.4：读LED状态

域	内容	含义	字节数
功能码域	1	操作LED灯	1
数据域	0	读取LED灯	1

域内容含义字节数

功能码域 1 操作LED灯 1

数据域 0 读取LED灯 1

ESP32发送读LED状态的帧给MSP430后，MSP430返回的帧如表 4.5所示

表 4.5：MSP430返回给ESP32的表示LED灯状态的帧

域	内容	含义	字节数
功能码域	1	LED灯状态	1
数据域	0或非零	0表示LED灯为灭，非零表示为亮	1

域内容含义字节数

功能码域 1 LED灯状态 1

数据域 0或非零 0表示LED灯为灭，非零表示为亮 1

MSP430发给ESP32的从DS18B20采集到的温度信息的帧格式如表 4.6所示。

表 4.6: MSP430发给ESP32的温度信息

域	内容	含义	字节数
功能码域	2	温度信息	1
数据域	float数据类型的温度	温度	4

域内容含义字节数

功能码域 2 温度信息 1

数据域 float数据类型的温度温度 4

1.5.2 MQTT协议简介

MQTT是ISO 标准(ISO/IEC PRF 20922)下基于发布/订阅范式的消息协议。它工作在 TCP/IP协议族上，是为硬件性能低下的远程设备以及网络状况糟糕的情况下而设计的发布/订阅型消息协议。[7] MQTT的订阅/发布模型示意图如图 4.1所示

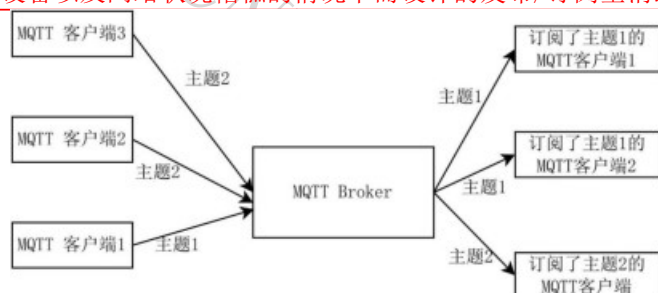


图 4.1: MQTT的订阅/发布模型示意图

MQTT客户端1发布了主题1，有两个客户端订阅了主题1，这两个客户端都能收到主题1的消息。MQTT客户端2和MQTT客户端3都发布了主题2的消息，订阅了主题2的MQTT客户端可以在任何一个MQTT客户端发布了主题2的消息时收到消息。即MQTT协议可以实现一对一、一对多、多对一的通信。一个MQTT客户端可以同时是订阅者和发布者。

MQTT面向性能低下的物联网设备，它使物联网设备不用进行差错控制和重传，一对多通信时也不用物联网设备对每个接收方分别发送一次，只需要发送一次给MQTT Broker，由MQTT Broker负责消息的分发、差错控制、接收确认、重传等。MQTT Broker可以工作于多种传输层协议上，例如TCP，UDP，WebSocket 等。由于浏览器的限制，运行在浏览器上的MQTT协议无法使用TCP或UDP协议，只能使用WebSocket协议。NuGet中的 .NET 的MQTT库不支持在Blazor环境下运行。Blazor可以使C# 和JavaScript互操作，而JavaScript有开源的基于WebSocket的MQTT库mqtt.js，可以通过C# 和JavaScript互操作来编写MQTT客户端组件。

通过MQTT，还可以解决消息边界定位的问题，MQTT每个消息都会给出载荷的长度，MQTT支持发送长报文，可以根据设备的内存大小定义最大报文大小，最大可以达到256MB而无需在发送方进行分割，接收方进行拼接，由MQTT协议自动完成。对于物联网设备，无需传输太长的报文，远小于256MB。阿里云等企业有高性能的MQTT Broker出租，也有一些企业提供免费的MQTT Broker服务，但是性能和安全性较低。也可以使用开源的，免费的MQTT Broker软件部署私有的MQTT Broker。

1.5.3 对MQTT客户端ID格式的规定

下文中出现的像 [xxxx] 这种形式的，方括号里面包含着一些文字的，这里约定为：把方括号和里面的文字整体替换为方括号内的文字描述的内容。

每个 MQTT 客户端都有一个客户端 ID，是字符串的形式。在同一个 MQTT 服务器中，客户端 ID 不能一样。

控制器的客户端ID格式为：[设备类型]-[MAC地址]，例如本项目中的温度控制器使用esp32作为设备类型名，则其客户端ID格式为：esp32-[MAC地址]。每个控制器只会建立一个MQTT连接。

上位机的客户端 ID 为：[用户名]-[32位16进制随机数]。其中，随机数生成使用的是C# 的Guid类，该类能够根据能够获得的硬件信息、时间等作为种子，计算出一个128位二进制的随机数，能够保证不会重复。使用随机数的目的是为了上位机能够同时建立多个MQTT连接，也是为了让一个用户能够使用多个上位机与控制器进行通信。

1.5.4 对MQTT主题的规划

为了让MQTT客户端之间能够实现与特定的对象的一对一、一对多、多对一通信，也为了实现本项目中的使系统具有良好的灵活性和可拓展性的目标，需要对MQTT的主题做出一些规划。

1.5.4.1 子主题

MQTT的主题可以使用斜杠 / 来划分子主题，形式上类似URL。例如：esp32/Temperature 就是一个有两个子主题的主题。MQTT客户端在订阅主题时可以使用通配符。通配符有 # 和 +，其中，+ 指代一个子主题，# 指代任意数量的子主题。

1.5.4.2 主场、客场

声明：“主场”、“客场”这两个词不是标准内的定义，只是本文为了方便讲述而取的名称。

将主题的第一个子主题称为场。第一个子主题的内容是MQTT客户端的客户端ID。例如主题“esp32-7C:9E:BD:65:DA:E0/out/Temperature”中的第一个子主题是一个温度控制器的MQTT客户端ID。

如果一个MQTT客户端订阅或发布了一个主题，该主题的场是它自己的客户端ID，则称该设备订阅发布的方式为主场，如果是其他MQTT客户端的客户端ID，则称为客场。于是有了“主场发布”、“客场发布”、“主场接收”、“客场接收”这几个概

念。

1.5.4.3 主场发布

例如 esp32-7C:9E:BD:65:DA:E0 发布了如下主题

esp32-7C:9E:BD:65:DA:E0/out/Temperature

该主题的一级子主题是它自己的客户端ID，所以是主场发布。

主场发布的消息的接收方采用的是客场接收。主场发布不针对特定的客户端，发送方并不知道有多少个接收方，是一对多的发布，任何订阅了该主题的MQTT客户端都可以接收到消息。

1.5.4.4 客场发布

例如上位机想要控制 esp32-7C:9E:BD:65:DA:E0 的 LED 灯，于是对着如下主题发布了一条消息

esp32-7C:9E:BD:65:DA:E0/in/LED

因为该主题的一级子主题是一个温度控制器的客户端ID，而不是上位机的客户端ID，所以上位机的这种发布方式称为客场发布。

客场发布的主题的接收者采用的是主场接收。客场发布是多对一的发布，任何一个知道了接收者的客户端ID的发布者都可以对该接收者发送消息。如果需要告诉接收者该条消息的发送者是谁，可以在载荷中带上自己的客户端ID。

1.5.4.5 主场接收

例如一个温度控制器的客户端ID是esp32-7C:9E:BD:65:DA:E0，它接收到了如下主题的消息

esp32-7C:9E:BD:65:DA:E0/in/SetLEDState

该主题的第一个子主题是该温度控制器的客户端ID，所以是主场接收。而发布该主题的上位机采用的方式就是客场发布。

1.5.4.6 客场接收

例如，上位机接收到来自温度控制器esp32-7C:9E:BD:65:DA:E0的如下主题的消息。

esp32-7C:9E:BD:65:DA:E0/out/Temperature

该主题的一级子主题不是上位机的客户端ID而是温度控制器esp32-7C:9E:BD:65:DA:E0的客户端ID，所以，对上位机来说是客场接收。而对于控制器esp32-7C:9E:BD:65:DA:E0来说就是主场发布。

1.5.5 上位机和控制器之间的通信

上位机和控制器之间使用MQTT协议进行通信，而双方想要进行通信，就需要一方订阅特定的主题，另一方发送该主题的消息。在4.4 中定义了“主场”，“客场”的发布/接收方式，下面，将介绍上位机和控制器之间如何通过4.4 中定义的方式进行通信以及为什么要使用这种方式。

1.5.5.1 控制器发送消息的方式

控制器发送消息时采用主场发布。采用主场发布的优点有：

1. 如果一个上位机对ESP32发送了命令，控制器执行命令后状态发生了改变，需要发送响应消息给上位机，告诉上位机状态已发生改变，采用主场发布使得所有上位机都能够收到该响应消息，从而获知控制器状态已改变。
2. 控制器使用主场发布使得控制器无需维护一个上位机的列表，在需要发送数据时遍历每个节点，分别对每个上位机发送消息。这对性能低下的控制器是一个负担。
3. 控制器返回的响应消息的载荷中无需附带控制器的客户端ID，因为上位机需要同时与多个控制器通信，如果控制器发送消息采用客场发布，主题中不含控制器的信息，上位机无法确定该消息来自哪个控制器，所以，控制器需要在响应消息中携带控制器自己的客户端ID，这增加了程序的复杂度，同时控制器需要发送的消息体更长了，更消耗性能。

1.5.5.2 控制器接收消息的方式

控制器接收消息采用主场接收。采用主场接收的优点如下：

1. 控制器无需维护一个上位机的列表，订阅所有上位机的主题，以在上位机发送消息时自己能够收到。
2. 控制器无需随时关注新的上位机上线或有上位机下线了，从而更新自己订阅的主题。控制器只需在连接到MQTT服务器时使用通配符订阅一个主题，该主题为：
[控制器的客户端ID]/#
即使用通配符订阅自己主场的所有主题。这使得控制器初始化时执行的工作变简单，使性能低下的控制器初始化时间能够较短。

3. HTTP服务器无需提供获取当前在线上位机的接口。如果控制器需要能够获取当前在线的上位机，需要在HTTP服务器中提供接口，让控制器通过HTTP请求获取当前在线的上位机列表。这使得控制器和HTTP服务器的程序更复杂。同时，控制器在维护MQTT连接的同时又要使用HTTP协议，增加了负担。

1.5.5.3 上位机发送消息的方式

上位机使用客场发布的方式发送消息。原因如下：

1. 上位机是性能较强的计算机，有能力维护一个控制器的列表，在需要对所有控制器发送消息时遍历该列表中的每一个控制器。
2. 控制器难以采用客场接收，只能采用主场接收，所以上位机必须使用客场发布。

1.5.5.4 上位机接收消息的方式

上位机采用客场接收的方式接收消息。原因如下：

1. 使用客场接收，主题中携带了控制器的客户端ID的信息，上位机无需要求控制器在消息的载荷中携带控制器的客户端ID，主题采用的是字符串，而客户端ID采用的也是字符串，上位机能够通过分隔符轻易地提取出控制器的客户端ID，而在载荷中携带客户端ID，上位机还需要确定出该客户端ID的消息边界，提高了程序的复杂度。

2. 控制器只能采用主场发布

1.5.6 本章小结

本章介绍了MSP430和ESP32之间UART通信所适用的自定义的协议，接着介绍了MQTT协议，对MQTT客户端的客户端ID格式做出了规定，并对不同类型的客户端能够订阅/发布的主题做出了限制，这是安全认证和本系统各部分能够有序通信的基础，也是本

系统灵活性和可扩展性的基础。通过对MQTT的客户端ID和主题的规划，使得所有遵循该规范的上位机和控制器之间能够使用相同的方式进行通信，达到了统一接口的目标。

上位机采用客场发布、客场接收；控制器采用主场发布、主场接收。一方面是因为控制器性能低下，难以胜任使用客场接收，客场发布。另一方面，因为上位机的客户端ID中含有随机生成的部分，而控制器的客户端ID是固定的。维护控制器的客户端ID列表比维护上位机的客户端ID列表容易，这可以简化程序的设计，同时，也给性能较低的控制器减小了负担。

通信协议设计

1.6 系统软件设计

软件设计涉及到：网页客户端的设计、ESP32和MSP430通信程序的设计、对MQTT协议的使用、HTTP服务器的设计。还涉及到了安全认证。最后对系统的通信功能进行了展示。

1.6.1 采用的平台或框架

MSP430的软件设计采用Energia框架。Energia 是一个开源和社区驱动型集成开发环境（IDE）与软件框架。为微控制器编程提供了直观的编码环境和由易于使用的功能 API 及库构成的可靠框架。Energia 支持多种 TI 处理器，主要包括可从 LaunchPad 开发生态系统获得的处理器。[6]

ESP32的软件设计采用Arduino框架。该框架将FreeRTOS移植到ESP32中。早期的嵌入式开发采用基于裸机的前后台系统。前台程序为中断服务程序，用来设置标志位，后台程序为死循环，用来轮询标志位。这种系统的实时性比预计的差。因为所有任务都具有相同的优先级，更紧急的任务无法先得到处理。[7]

HTTP服务器采用ASP.NET Core平台。ASP.NET Core 是一个跨平台，高性能，开源的框架，用于构建现代的，支持云计算的，联网的应用。[8]

网页端采用 Blazor 框架进行开发。Blazor是一个使用 .NET构建客户端的网页UI的框架。[9]

1.6.2 控制器的接口实现规范

控制器使用主场接收的方式接收控制命令。例如，本项目中的温度控制器订阅了主题：[控制器的客户端ID]/in/#，# 是通配符，上位机想要控制MSP430上的一个LED灯的亮灭，就发布一个主题为 [控制器的客户端ID]/in/set_LED_state 的消息，在载荷里附带对LED状态的控制命令。

控制器采用主场发布的方式发布消息。例如，控制器需要将MSP430的LED灯的状态发送给上位机，就发布一个主题为 [esp32的客户端ID]/out/LedState 的消息，在载荷中附带LED灯的状态信息。

站在控制器的角度，接收数据全部在二级子主题 in 下面，发送数据全部在二级子主题 out 下面。从第三个子主题开始就是与功能相关的。例如，设置LED灯的主题为 set_LED_state 。采用字符串的形式来表示对应的功能对于人类友好，阅读和调试方便，定义主题的时候也方便。

in 子主题下面的子主题采用下划线分隔每一个单词，首字母小写。out子主题下面的子主题采用大驼峰命名法，例如 LedState，每个单词的首字母大写。

1.6.3 减小MQTT主题匹配的时间复杂度

采用主题来表示功能虽然对人类友好，但是，对于计算机，查找字符串的速度没有直接使用整型变量通过switch-case随机访问的速度快。使用整型变量表示功能的方式是ModBus等协议所使用的方式。使用这种方式只需对功能码使用switch-case语句进入相应的处理函数就行，但是要求程序员编程时查阅文档，且记忆困难，容易出错。使用MQTT协议后，如果只使用一个主题，而通过在载荷中附带功能码，使用类似ModBus的那种方式，这就违背了MQTT的设计初衷。ESP32接收到消息后，通过斜杠 / 将子主题分隔出来，放到 string 对象数组里，第三个子主题是代表功能的，数量会比较多，要求快速查找第三个子主题然后进入相应的处理函数。如果采用 if-else if 嵌套的结构来查找，时间复杂度是 $O(n)$ ，最坏的情况需要对比到最后一个字符串才能进行相应的工作。而 C++ 不像C#或JAVA等语言支持对字符串的look up switch，所以，需要程序员手动使用 STL 的 map类来建立查找表。

可以在初始化时建立map对象，键为储存第三个子主题的string对象，值为std::function对象，使用lambda表达式来为节点赋值。例如图 5.1所示

```
m_commandMap["read_LED_state"] = [](uint8_t *payload, uint32_t length) -> void
{
    uint8_t readLedState = 0;
    pTar->sendData(1, &readLedState, 1);
};
```

图 5.1：初始化第三个子主题的map节点

然后在接收到一条消息后，分割出第三个子主题，查找map，调用对应的函数。如图 5.2所示

```
auto it = m_commandMap.find(subTopics.getItemAtIndex(index++));
if (it != m_commandMap.end()) //如果在map中找到处理该子主题的函数
{
    (it->second)(payload, length);
}
```

图 5.2：查找map调用对应的函数

通过这种方式，查找主题的时间复杂度为 $O(\log n)$ 。

此外，使用这种方式使得程序的结构清晰明了。需要实现一个新的功能接口时，只需要使用图 5.1中的方式，在map中添加一个节点即可。这带来了良好的可扩展性。与ModBus使用数字的功能码相比，程序可读性也大大提高，与URL一样实现了所见即所得。

此外，使用MQTT进行通信也可以兼容功能码的方式。如果某款控制器对实时性要求较高，可以减少3级子主题的数量，在MQTT消息载荷中使用功能码+数据的格式，利用switch-case语句根据功能码跳转到相应的处理函数。

1.6.4 HTTP服务器的接口和安全

HTTP服务器提供了以下接口：

- 1. 登录认证接口
- 2. MQTT连接认证接口
- 3. MQTT订阅、发布认证接口
- 4. 获取设备信息接口

下面将分别对这些接口进行介绍。

1.6.4.1 登录认证接口

“登录认证接口”用来给客户端提供用户登录认证的功能。客户端在用户输入用户名、密码，并点击登录按钮后，通过HTTP的POST请求的方式将用户名、密码放在请求体中发送给HTTP服务器。HTTP服务器从请求体中获取用户名、密码后，查询数据库，看是否存在一条记录的用户名、密码与本次请求中的用户名、密码一致，如果找到，证明用户名、密码正确，通过认证，否则不通过认证。如果通过认证，则HTTP服务器会返回200状态码，不通过认证则返回404状态码。

在数据库中建立用户表，储存用户的用户名和密码。每个用户的用户名用来确定用户的身份，必须唯一。用户表如图5.3所示：

	username	password
1	hjc	123456

图 5.3：用户表

登录认证接口的程序流程如图 5.4所示：

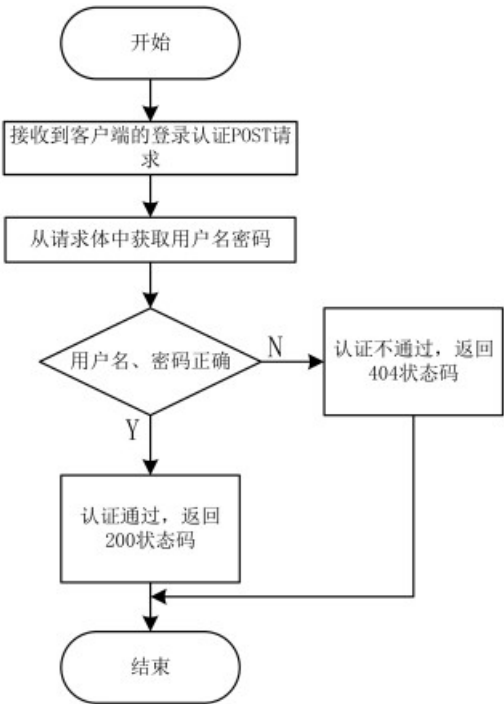


图 5.4：登录认证接口

1.6.4.2 MQTT协议的安全性

MQTT协议本身支持使用用户名、密码进行认证，但是，如果仅使用这种认证方式，仍然存在着一些安全问题。

1.6.4.2.1 安全问题1：恶意抢占其他MQTT客户端的连接

任何拥有账号的用户都能用自己的用户名、密码连接到MQTT 服务器，且客户端ID是由MQTT客户端自己决定的。这就会产生一个问题：如果一个恶意的用户注册了一个账号，并且获取到其他MQTT客户端的客户端ID，就可以使用自己的用户名、密码和该MQTT客户端的客户端ID向MQTT 服务器发起连接请求，如果仅有用户名、密码的验证，经过检查，发现该连接的用户名、密码都是正确的，MQTT服务器就通过该连接请求，这就会导致另一个正常的，也使用该MQTT客户端ID的用户的连接被抢占，因为MQTT 服务器不允许重复的客户端ID，如果有重复的客户端ID，新的连接会覆盖旧的连接。

例如，使用MQTTfx（一个图形界面的MQTT客户端软件），打开两个窗口，两次使用同一个客户端ID连接到同一个MQTT服务器，并且用户名、密码都是正确的，第一个窗口先连接，第二个窗口再进行连接，在第二个窗口连接成功后，第一个窗口出现了如图 5.5所示的提示。

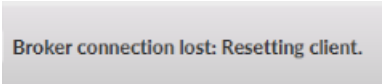


图 5.5：第一个窗口出现的提示

该提示说明第二个窗口的连接抢占了第一个窗口的连接。

1.6.4.2.2 安全问题2：窃听

一个MQTT客户端通过连接认证后，由于MQTT协议本身没有对订阅和发布做出规范和限制，所以有可能存在一个恶意的MQTT客户端在通过连接认证后订阅所有其他MQTT客户端的消息来窃取消息，或者对其他MQTT客户端频繁发送消息进行DOS攻击，或者通过发送消息来控制其他MQTT客户端，以本系统为例，可能恶意将温度控制器的给定温度调高，从而导致用户出现经济损失。

1.6.4.2.3 解决方案

要解决安全问题1，必须对MQTT客户端连接时使用的客户端ID进行检查。要解决安全问题2，必须对MQTT客户端能够订阅/发

布的主题进行限制。

解决这两个安全问题离不开4.3节中对MQTT客户端ID进行的规范，离不开4.4节中对MQTT主题的规划。所有的MQTT客户端都必须遵守这两个规范才能通过验证。

针对这两个安全问题，HTTP服务器实现了“MQTT连接认证接口”和“MQTT订阅、发布认证接口”。

本项目使用emqx搭建MQTT服务器。emqx是一个开源的，免费的MQTT服务器软件。emqx除了提供MQTT Broker（MQTT代理）的功能外，还支持一些插件。安装HTTP认证插件，在配置文件中设置认证方式为HTTP认证，将请求的地址设置为HTTP服务器所在的地址。当有设备请求连接到MQTT服务器或想要订阅和发布主题时，MQTT服务器会向HTTP服务器发送POST请求，请求体中携带MQTT客户端的信息和MQTT客户端的行为信息（行为信息为：MQTT客户端的行为是连接还是订阅/发布，订阅或发布的主题是什么，连接所使用的用户名密码等），HTTP服务器可以通过返回200状态码来表示通过认证，通过返回404状态码表示认证不通过。

通过配置emqx就可以使用“MQTT连接认证接口”和“MQTT订阅、发布认证接口”了。这两个接口的机制可以总结为一张流程图，如图 5.6所示。

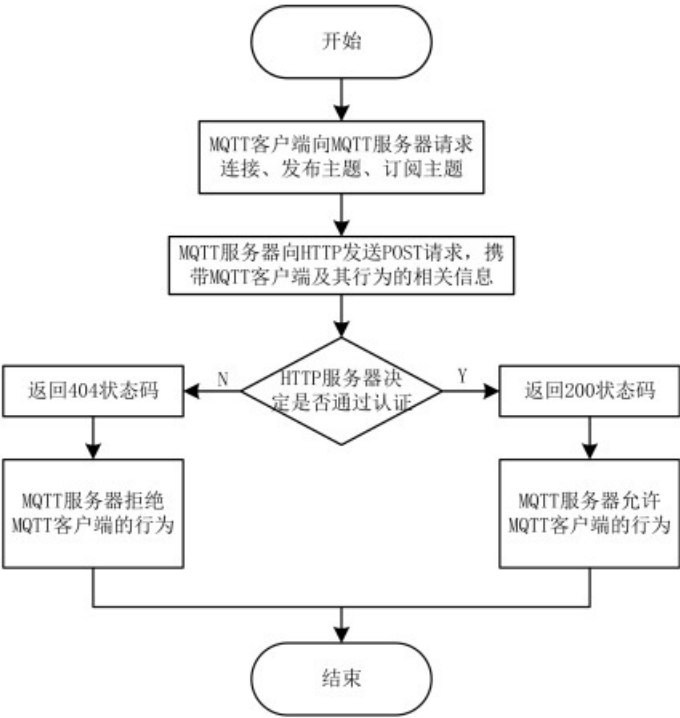


图 5.6：MQTT安全认证

通过使用“MQTT连接认证接口”和“MQTT订阅、发布认证接口”，再配合使用加密算法对数据的传输进行加密，即可保障MQTT协议的安全。下面，将分别介绍这两个接口。

1.6.4.3 MQTT连接认证接口

MQTT连接认证接口主要针对安全问题1.

1.6.4.3.1 接口的工作原理

根据4.3节中对客户端ID格式做出的规定，上位机的客户端ID格式为：[用户名]-[32位16进制随机数]，控制器的客户端ID格式为：[设备类型]-[MAC地址]，HTTP服务器首先根据客户端ID的格式判断出该MQTT客户端是来自上位机还是来自网页，接着对两者做出不同的限制。

当上位机连接到MQTT服务器时，检查其客户端ID中的[用户名]，规定：当上位机以用户A的身份连接到MQTT服务器时（连接时使用的用户名是用户A即为以用户A的身份），[用户名]必须是用户A。在对客户端ID的合法性进行检查后，还需要验证用户名、密码的正确性。

控制器的连接到MQTT服务器时所使用客户端ID也是储存在数据库中的该控制器的唯一ID。在数据库中建立了设备表，其中有两个字段，分别是device_id和username，分别为设备的ID和它所属的用户。如图 5.7所示：

device_id	username
esp32-7C:9E:BD:65:DA:E0	hjc
esp32-test	hjc
test-1	hjc

图 5.7：设备表

当控制器连接到MQTT服务器时，如果使用的用户名、密码是用户A的用户名、密码，则检查其所使用的客户端ID，具体为：查询数据库，用户A名下是否有设备ID为该客户端ID的控制器，如果有，进一步检查用户名、密码的正确性。

指 标

疑似剽窃文字表述

1. 为微控制器编程提供了直观的编码环境和由易于使用的功能 API 及库构成的可靠框架。

相似文献列表

去除本人文献复制比：2.5%(219)		文字复制比：2.5%(219)	疑似剽窃观点：(0)
1	基于远程控制的SD卡文件管理		2.1%（182）
	惠旦(导师：曾培峰) - 《东华大学硕士论文》 - 2012-03-01		是否引证：否
2	12785431_王保阳_个性化推荐酒店管理系统应用与研究		0.4%（35）
	王保阳 - 《高职高专院校联合比对库》 - 2018-05-18		是否引证：否

原文内容

全部检查都通过后才通过认证，否则不通过。

通过上述的对控制器和上位机连接到MQTT服务器时所适用的客户端ID、用户名、密码做出检查，即可解决安全问题1，除非用户的用户名和密码泄露，否则恶意用户无法抢占正常用户的连接。

1.6.4.3.2 接口的程序设计

根据需求，通过配置文件设置当emqx收到MQTT客户端的连接请求后向HTTP服务器发送POST请求体的JSON中包含哪些信息。本项目中设置JSON包含的键有：clientid, username, password. 分别是客户端ID，用户名，密码

MQTT连接认证的程序如图 5.8所示：

图 5.8：MQTT连接认证流程图

为了节约性能，认证遵循从简单、快速的到复杂、费时的原则。先从简单的客户端ID的格式判断开始，该认证不需要连接数据库，能够先筛选出客户端ID不合格的。接着进行较费时的需要连接数据库的认证。

1.6.4.4 MQTT订阅和发布认证接口

MQTT连接认证接口主要针对安全问题2.

1.6.4.4.1 接口的工作原理

对于控制器，因为限制了控制器只能主场接收、主场发布，所以控制器是处于被动状态的，如果其他MQTT客户端不主动订阅该控制器的场的主题，是收不到来自该控制器的消息的。当一个来自控制器的MQTT客户端请求订阅/发布主题时，只需判断是否主场即可。

上位机是用来控制控制器的，与控制器通信使用的是客场接收、客场发布，处于主动状态，它的订阅/发布行为会对其他MQTT客户端产生影响，此外，上位机存在随意订阅其他MQTT客户端的主题从而窃听消息的可能，所以必须做出限制。规定上位机以用户A的身份连接到MQTT服务器后，只能对用户A所属的控制器进行客场接收和客场发布。这种限制即防止了恶意用户控制他人的控制器，也防止了消息的窃听。

1.6.4.4.2 接口程序设计

图 5.9：订阅/发布认证流程

MQTT客户端进行订阅/发布时，MQTT服务器emqx向HTTP发送的POST请求体使用JSON承载信息。可以通过配置文件设计JSON中包含哪些信息。在本项目中设置该JSON包含的键为：clientid, username, topic. 订阅/发布认证的流程图如图 5.9所示。

1.6.4.5 获取设备信息接口

上位机使用的是客场接收、客场发布的方式与控制器进行通信，因此上位机想要与控制器进行通信必须先知道控制器的客户端ID，可以通过用户手动输入或上位机自己使用此接口来获取这两种方式来实现。

此接口的实现原理为：上位机通过HTTP的POST请求，将用户名、密码发送给HTTP服务器，HTTP服务器查询数据库中的用户表，验证用户名、密码，验证通过后查询设备表，将该用户所拥有的设备的ID以HTTP响应的形式，放在响应体中送回给请求者。如果用户名、密码验证不通过，则HTTP服务器返回404状态码。该接口的流程图如图 5.10所示。

图 5.10：获取设备信息接口

1.6.5 网页端用户登录认证

网页端用户登录用到了HTTP服务器的“登陆认证接口”。网页端的用户登录界面如图 5.11所示。如果登录失败会在上方出现弹窗提示，3秒后消失。

图 5.11：登录界面

网页客户端使用“登陆认证接口”进行用户登录的流程如图 5.12所示。

图 5.12：网页客户端用户登录流程

用户登录成功后会通过Blazor的路由跳转到用户设备列表界面。在该界面会呈现用户所拥有的控制器的控制面板。获取用户所拥有的控制器需要用到HTTP服务器提供的“上位机获取用户所持有的控制器的信息的接口”。

网页登录成功后的每一个操作，例如获取控制器信息、通过MQTT与控制器进行通信，都需要经过HTTP服务器的认证。这种方式 and 超星学习通的人脸识别认证不同。超星学习通的人脸识别认证只在特定的页面会要求用户进行认证，用户认证成功后会跳转到另一个页面。如果用户将跳转后的页面的URL记住，下次直接使用该URL进行访问，就可以不进行验证直接进入该页面。

本项目所使用的认证机制杜绝了这种可能。用户无法通过认证成功后Blazor路由跳转到的页面的URL绕过认证直接进入该页面。一方面是因为Blazor的机制。所有路由地址的URL在服务器端都不对应一个真正路径，直接访问浏览器会提示找不到资源。

另一方面，在本项目中，为了不让用户访问这些URL时浏览器显示找不到资源的页面降低用户体验，同时为了引导用户到正确的页面，HTTP服务器会将对非静态文件资源，且没有被其他终结点中间件处理的URL全部重定向到根路径。如图 5.13所示。

该中间件需要紧跟着静态文件中间件并位于静态文件中间件之前，否则会令其它所有中间件（例如路由中间件）失效。静态文件中间件是一个终结点，所以必须位于静态文件中间件之前，在进入静态文件中间件之前把所有非静态文件的URL进行重定向，否则，进入静态文件中间件后，文件系统找不到文件会返回404状态码。

图 5.13: 重定向

HTTP服务器只有静态文件资源是不需要认证直接可以获取的，除此以外的所有接口的使用都需要通过认证。

1.6.6 设备列表组件

设备列表组件的界面是用户登录成功后进入的界面。设备列表组件的界面如图 5.14所示。

图 5.14: 设备列表界面

该界面中每个方框内都是一个控制器的控制面板。在图 5.14中有3个控制器的控制面板。从控制面板可以看出，esp32-7C:9E:BD:65:DA:E0和esp32-test是同类型的控制器，而test-1是另外一种类型的控制器。

设备列表组件在初始化时，会使用HTTP服务器的“获取设备信息接口”，获取当前登录用户所拥有的控制器的信息，根据这些信息加载出每一个控制器的控制面板。

该组件在获取到用户的控制器信息后，将这些信息存放到名为 `_deviceList` 的列表中，该列表为字符串类型的列表，其中的设备信息为每一个设备的ID的字符串。razor组件的编程语法支持C# 和html 语法相互嵌套来动态地构建页面。如图 5.15所示

使用razor语法，遍历 `_deviceList` 中的每一个设备，提取出设备ID的前缀，以此判断该设备的类型，根据不同的设备类型加载不同的组件。例如：前缀为esp32则加载ESP32组件，前缀为test则加载TestDevice组件。ESP32组件和TestDevice组件的界面即为图 5.14中所呈现的控制器的控制面板。

图 5.14呈现控制面板的控制器是图 5.7中所示的数据库中的设备表所储存的控制器。

所有的控制面板组件，在被加载之前都需要传递参数进去，如图 5.15，在加载组件前传递了ClientId和Initialized两个参数。其中，ClientId为控制器的ID，控制器的ID也是4.4 节中定义的控制器订阅和发布的主题的场，传递该参数给控制面板组件后，组件内部就能通过客场订阅、客场发布的方式和该控制器进行通信。

图 5.15: C# 和html互相嵌套的语法

通过razor语法，再配合4.3 中对MQTT客户端格式的规定，即可实现加载出不同类型的控制器的控制面板。当新出现了一种控制器，需要在网页中呈现这种控制器的面板时，只需要设计该控制器的控制面板组件，然后在设备列表组件的switch-case表中添加对这种控制器的类型名的跳转分支即可。

1.6.7 ESP32组件

ESP32组件用来提供对esp32类型的控制器的控制面板。该组件的界面如图 5.16所示：

图 5.16: ESP32组件界面

下面分别介绍该组建的功能。

1.6.7.1 在线状态显示

该组件能够实时显示设备是否在线。该组件订阅了对应的ESP32的主题：`[ESP32的客户端ID]/out/IsOnline`。该主题用来获取ESP32的在线状态。ESP32会通过两种方式发送该主题的消息。一种是收到来自网页的询问是否在线的消息后主动发送的响应消息；另一种是ESP32的遗嘱。遗嘱是MQTT的一种机制。每个MQTT客户端都通过心跳向 MQTT Broker 表明自己在线，如果MQTT客户端设置了遗嘱，在MQTT客户端非正常断开连接时（即没有通过握手断开连接，MQTT Broker 却收不到MQTT客户端的心跳了）MQTT Broker会对订阅了该客户端的遗嘱主题的其他客户端发送该客户端的遗嘱。MQTT Broker会在超过MQTT客户端设置的心跳时间的两倍后仍然没有收到心跳包的情况下认为MQTT客户端意外断线，这个机制是为了防止网络延迟和MQTT客户端恰好在忙于其他任务导致发送心跳包的时间间隔有偏差。遗嘱是在与MQTT Broker建立连接时设置的，遗嘱和普通的MQTT消息一样，也可以设置一个主题。ESP32的遗嘱主题为：`[ESP32的客户端ID]/out/IsOnline`。另外，在ESP32收到询问是否在线的消息后，也会向该主题发布消息。如图 5.17所示，ESP32离线后网页会更新指示灯。

图 5.17: ESP32离线后更新指示灯

1.6.7.2 温度显示

该组件还能够显示MSP430连接的DS18B20采集的温度信息。MSP430设置的温度采样时间间隔较长，长达5秒。如果用户打开网页后长时间没有加载出温度信息，会大大降低用户体验。因此，ESP32在建立与MQTT Broker的连接时，将干净会话选项设置为false。该选项用来设置MQTT Broker是否保留MQTT客户端发布的消息，即MQTT客户端每发布一个消息，MQTT Broker都会将该消息储存起来。当MQTT客户端再次发布一条同主题的消息，MQTT Broker会用新的消息覆盖旧的消息。连接时设置的干净会话选项会成为之后的默认选项，在发布每一条消息时还可以单独设置是否为干净会话。设置为非干净会话后，只要有MQTT客户端订阅了一个主题，该主题恰好有储存在MQTT Broker中的副本，MQTT Broker就会将已储存的副本立刻发送给订阅者。通过这种方式，能够使网页快速获得温度信息，无需等待MSP430的下一次采样完成。

1.6.7.3 LED开关

该开关采用闭环控制。ESP32组件初始化时，会发送消息给ESP32，读取MSP430的红色LED灯的状态。ESP32收到消息后，通过UART发送消息给MSP430，MSP430返回当前LED灯的状态，再由ESP32通过MQTT发送给网页。网页获取到信息后更新开关的状态。用户点击按钮后，网页发送消息给ESP32，要求设置MSP430的LED灯状态。ESP32收到消息后通过UART发送给MSP430，MSP430将LED灯设置为指定的状态。ESP32还会发送一条消息，要求读取MSP430的LED灯状态，MSP430返回回答消息后，ESP32转发给网页，网页收到消息后更新开关的状态，让用户感知到开关LED灯的操作是否成功。如果没有收到反馈信息，LED灯开关的状态不会更新。控制的逻辑如图 5.18所示。

图 5.18: LED开关的控制逻辑

1.6.7.4 日志显示

日志显示的主题为：`[ESP32的客户端ID]/out/Log`。该主题用于给ESP32提供向网页输出日志的途径。

1.6.8 TestDevice组件

该组件是用来进行通信测试的，不是一个实际的控制器。该组件可以用MQTT通信的调试工具MQTTfx进行模拟。

该组件的界面如图 5.19所示。

图 5.19: TestDevice组件界面

下面介绍此组件的功能。

1.6.8.1 在线状态指示灯

和ESP32组件界面中的在线状态指示灯一样，这个指示灯也是用来指示该控制器的在线状态的。该组件的在线状态显示是通过订阅 [TestDevice的客户端ID]/out/IsOnline 主题来获取控制器的在线状态的。在TestDevice组件初始化时，会发布一条消息给TestDevice，询问控制器的在线状态，默认控制器不在线，只有收到控制器的回复后才会将该组件的类内部的标志在线状态的字段设置为在线状态，并更新界面中的指示灯。此外，[TestDevice的客户端ID]/out/IsOnline 主题也是TestDevice对应的控制器的遗嘱主题，在控制器离线后，MQTT Broker会发送控制器的遗嘱信息，网页收到遗嘱后就可以获知该控制器已离线。

1.6.8.2 日志显示

与ESP32组件的日志显示功能相同，用于将来自控制器的日志主题的消息的内容显示到网页中。

1.6.8.3 发送按钮

这个按钮被点击后就会向 [控制器的客户端ID]/in/hello 主题发送一条消息，方式为客场发布，控制器会收到这条消息。消息的内容为字符串“hello, I am Web”。

1.6.9 通信功能测试

1.6.9.1 以test-1的身份与网页进行交互

test-1是一个虚拟的控制器，对应的控制面板组件是TestDevice，下面，使用MQTTfx软件以test-1的身份与网页进行交互。MQTTfx是一个图形界面的MQTT通信调试软件。首先，建立与MQTT Broker的连接。用户名为 hjc 密码为 123456. 如图 5.21所示。

接下来需要设置遗嘱，以在关闭连接时能够更新网页上的在线状态指示灯。如图 5.20所示。因为该遗嘱较重要，需要设置服务质量等级为1，即保证接收方至少收到一次。遗嘱的主题为test-1/out/IsOnline，遗嘱的内容为字符串 false。

图 5.20: 设置遗嘱

图 5.21: 建立与MQTT Broker的连接

建立连接后，订阅主题test-1/in/#，这时控制器用来进行主场接收的主题。如图 5.22所示：

图 5.22: 订阅主题

接下来打开网页，并登录。网页中出现了如图 5.14所示的设备列表界面，其中呈现了每个控制器的控制面板。查看MQTTfx软件，显示收到了如图 5.23所示的消息。get_online_state为网页向testDevice发送的要求获取控制器在线状态的消息。

图 5.23: 收到的消息

此时，网页中的testDevice的在线状态指示灯为灰色。如图 5.24左测。

图 5.24: 控制器回复之前和回复之后的在线状态指示灯

接下来，使用MQTTfx发送消息，告诉网页自己在线。如图 5.25所示。

图 5.25: 告诉网页自己在线

消息主题为：test-1/out/IsOnline，内容为字符串 true，消息发出后可以看到网页中testDevice的在线状态指示灯亮起。如图 5.24右侧。

接下来向网页发送日志信息。日志信息采用的是字符串格式，如图 5.26所示：

图 5.26: 发送日志信息

消息发送出去后，网页中显示出了该条消息。如图 5.27所示

图 5.27: 网页中显示了日志信息

点击控制面板中的按钮，MQTTfx显示收到了如图 5.28所示的消息。这时网页发送的问候消息。

图 5.28: MQTTfx收到问候消息

消息的内容如图 5.29所示

图 5.29: 问候消息内容

接下来关闭MQTTfx的窗口，使得图 5.20中设置的遗嘱得以发送。遗嘱发送后，控制面板上的在线状态指示灯会由亮变成灭。亮的状态如图 5.30的(a)所示，灭的状态如图 5.30的(b)所示。

(a)

(b)

图 5.30: 关闭MQTTfx后在线状态指示灯的变化

1.6.9.2 以上位机的身份与控制器进行交互

本系统具有良好的可扩展性，可以方便地加入各种类型的上位机，所有的上位机和控制器进行通信的方式都相同。下面，不开启网页，以MQTTfx作为上位机，与控制器esp32-7C:9E:BD:65:DA:E0进行交互。

使用上位机的客户端ID格式，以用户hjc的身份连接到MQTT服务器。如图 5.31所示。连接成功后，订阅控制器esp32-7C:9E:BD:65:DA:E0的主场发布主题： esp32-7C:9E:BD:65:DA:E0/out/#，通过客场接收的方式接收来自控制器的消息。

订阅主题后，不断收到来自控制器的温度数据。如图 5.32所示：

图 5.31: 以上位机的身份连接到MQTT服务器

图 5.32: 收到控制器的温度信息

0000 0000 00A0 3C40 是16进制数，这是控制器发送的温度数据。温度数据是double类型的数，大小为8个字节，对应16位16进制数。

本次测试证明了本设计的系统具有良好的可扩展性，能够方便地加入各种类型的上位机，所有的上位机都能使用相同的方式和控制器进行通信。

1.6.10 本章小结

本章介绍了系统的软件设计。包括控制器的通信程序设计、HTTP服务器程序设计、网页程序设计，最终实现了一个使用网页客户端对控制器进行管理的物联网系统，网页端使用组件化设计，可以方便地添加新款控制器的控制面板。最后，使用MQTT通信调试工具MQTTfx分别模拟了上位机和控制器，证明了本系统的可扩展性和灵活性，能够方便地添加各种类型的上位机和控制器。所有上位机使用相同的方式和控制器通信，控制器使用相同方式实现接口。

软件设计

1.7 总结与展望

1.7.1 总结

在新时代乡村振兴的路上，需要大量人才为农业现代化做出贡献。当前农业现代化处于初级阶段，网络基础设施不完善，设备缺乏。本课题为自拟题，和导师协商后确定下来。本人有过人工孵化雏鸡的经历，因此决定以此为背景进行研究。

结合当前行业背景，各大智能家居企业使用的协议和接口不统一，互相不兼容，给用户带来了不便。在DCS发展的早期，各大厂商的设备也互相不兼容，后来出现了统一的标准，FCS得以发展。统一的标准和接口有助于生产力的发展，能够使用户自由选择价格合理且符合需求的来自不同厂家的产品，从而为用户节约成本。但是，DCS和FCS是面向具有一定的生产规模的用户，而我国现阶段的土地政策仍然是家庭联产承包责任制，大多数农户是个体户，乡村振兴离不开个体农户，农业信息化发展不能抛弃个体农户。因此，本项目设计了一个统一接口的，方便扩展的，具有很大灵活性的物联网系统，能够调整拓扑结构以满足不同生产规模的用户的需求，能够方便地加入各种类型的上位机。各种类型的控制器使用同样的接口与所有类型的上位机进行通信。

针对种蛋孵化对温度要求较严格，参考市场上已有的水床孵化箱，设计出一个包含了基于物联网的水箱温度控制。该系统的控制器采用模拟量控制，与市面上采用简单的根据回差温度带进行的开关量控制不同，使用PI控制能够应对较大的时滞，抗扰性能较好，温度较平滑，更能保证孵化质量。

1.7.2 展望

基于本项目中的物联网系统，可以使用复杂的控制算法。本设计中的物联网系统可以扩展各种用途的上位机，其中包括云计算服务器，通过部署云计算服务器，对控制器的复杂控制算法进行计算卸载，从而达到更好的控制效果，在网络出现故障时可以退化为普通的PI控制算法。由于能力有限，本项目中只对PI、模糊PI和两者与史密斯预估器的结合进行了控制算法的仿真。

由于条件的限制，本设计中水箱温度模型进行参数辨识使用的辨识模型与设计的模型有区别，参数辨识后也没有给出阶跃信号得出响应曲线进行模型的验证。在有条件时仍需要进一步研究。

本项目对安全性进行了研究，并实现了相关软件的设计，但是，MQTT本身支持使用加密算法对数据传输进行加密，本项目中没有使用。HTTP也没有使用加密算法进化为HTTPS。在安全性方面，仍有较大的进步空间。

本项目只实现了对MSP430的LED灯的控制，对于控制器的控制面板，仍有很多必要的功能没有实现，例如温度曲线的绘制、温度给定值的设定、PI参数的调整等，仍需进一步完善。

总结与展望

参考文献

- [1] 百度百科. 中共中央、国务院关于全面推进乡村振兴加快农业农村现代化的意见[DB/OL]. 百度百科, 2021. 03. 02/2022. 5. 24.
- [2] 湘湘带你看看社会. 乡村振兴主攻智慧农业，从七个方面全面突破！[DB/OL]. <https://baijiahao.baidu.com/s?id=1729079522555521558&wfr=spider&for=pc>, 2022. 04. 03/2022. 5. 24.
- [3] 农业农村部. “十四五”全国农业农村信息化发展规划[DB/OL]. <http://www.nahs.org.cn/gk/tz/202203/P020220310548074048506.pdf>, 2022. 2. 22/2022. 5. 24.
- [4] Q. Zhu, R. Wang, Q. Chen, Y. Liu and W. Qin, "IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things" [H]. 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2010, pp. 347-352, doi: 10.1109/EUC.2010.58.
- [5] 知乎, 孵化爱好者. 鸡蛋孵化所需要合适的温度和湿度[DB/OL]. https://zhuanlan.zhihu.com/p/114975190?ivk_sa=1024320u, 2020. 3. 1/2022. 5. 24.
- [6] 德州仪器. ENERGIA[DB/OL]. <https://www.ti.com/tool/cn/ENERGIA>, 2022. 5. 10/2022. 5. 24.
- [7] 温淑红. 嵌入式Linux系统原理——基于ARM Cortex-A8处理器[M]. 北京市海淀区学院路 37号: 北京航空航天大学出版社, 2014. 6: 86.
- [8] 微软. Overview to ASP.NET Core[DB/OL]. <https://docs.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-6.0>, 2022. 3. 26/2022. 5. 24.
- [9] 微软. ASP.NET Core Blazor[DB/OL]. <https://docs.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-6.0>, 2022. 4. 17/2022. 5. 24.
- [10] 李江, 左巍. 基于COBS串行通信协议的自适应最优帧长分析[J]. 计算机与现代化, 2013. 3:116-120.
- [11] 百度百科. MQTT[DB/OL]. <https://baike.baidu.com/item/MQTT/3618851?fr=aladdin>, 2021-12-13/2022. 5. 24.
- [12] 陈霆希, 杨余旺. 基于MQTT协议的自动化生产线数据可视化系统[J]. 兵工自动化, 2022, 41(02):5-10.
- [13] 姬丽雯, 高菊玲, 刘永华. 基于MQTT的草莓温室物联网监控系统设计[J]. 农业开发与装备, 2021(12):167-169.
- [14] 崔金涛, 佟瑞栋. 基于ESP32及MQTT协议的视觉追踪机器人设计[J]. 辽宁师专学报(自然科学版), 2021, 23(04):67-71.
- [15] 冯毅琨, 白玉鹏, 侯志松. 基于MQTT和无线传感网络的大气质量监测系统研究与开发[J]. 河南科技学院学报(自然科学版), 2021, 49(06):73-80.
- [16] IBM. MQ 遥测传输 (MQTT) V5.0 协议规范[S]. IBM developerWorks, 2019.
- [17] 彭松. 基于MQTT的物联网安全技术研究与应用[D]. 北京邮电大学, 2019.
- [18] 闫乃实. 个性化家居植物浇灌系统的设计与实现[D]. 北京邮电大学, 2021. DOI:10.26969/d.cnki.gbydu.2021.000372.

[19] 何非. 基于ARM的HDL C协议通信控制器设计与实现[J]. 信息与电脑(理论版), 2019(16):75-76+79.

[20] 代玉梅, 张彬, 张瑞玲. 高级数据链路控制(HDL C)的操作行为研究[J]. 现代电子技术, 2013, 36(08):10-12. DOI:10.16652/j.issn.1004-373x.2013.08.028.

[21] 微型机与应用, 德州仪器扩展物联网(IoT)云生态系统并通过开源Energia支持简化代码开发[J]. 2015, 34(01):19. DOI:10.19358/j.issn.1674-7720.2015.01.008.

[22] 田赛. 串口数据通信协议的问题分析[J]. 科技视界, 2013(23):30+91. DOI:10.19694/j.cnki.issn2095-2457.2013.23.021.

[23] 孙思杰. Web项目基于前后端分离模式的设计与应用[J]. 科技创新与应用, 2020(27):96-97.

[24] 余佳恕. 基于ASP.NET MVC与MVVM技术的小型网站框架开发及应用[J]. 科学技术创新, 2020(20):101-102.

[25] 王宇. 基于MVVM前后端分离的物联网维管系统的研究与实现[D]. 北京工业大学, 2019.

DOI:10.26935/d.cnki.gbjgu.2019.000174. (北京爱琴海乐之技术有限公司, 2018)

致谢
本设计是在指导老师朱敏老师的帮助下完成的。老师在设计过程中给了我很大帮助，特别是在文档撰写方面。本课题是自己选题，由于不清楚本科毕业设计应该达到什么水平，最开始的题目不完善，工作内容不是很明确，经过和老师多次协商，再加上本人的学习，最终确定了研究内容，对要做的工作也有了大致的规划。

开题报告和毕业设计说明书在老师的帮助下改正了很多表述不当，语义不清，结构不合理，内容不完善、命名不规范、格式不规范等错误。没有老师的帮助，我无法完成文档的撰写。

作者：黄锦灿
2022年5月24日

指 标	
疑似剽窃文字表述	
1.	软件设计
1.	
2.	总结与展望
致谢	
本设计是在指导老师朱敏老师的帮助下完成的。老师在设计过程中给了	

表格检测详细结果

说明：1. 总文字复制比：被检测论文总重合字数在总字数中所占的比例

2. 去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例

3. 去除本人文献复制比：去除作者本人文献后，计算出来的重合字数在总字数中所占的比例

4. 单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比

5. 复制比：按照“四舍五入”规则，保留1位小数

6. 指标是由系统根据《学术论文不端行为的界定标准》自动生成的

7. 红色文字表示文字复制部分；绿色文字表示引用部分；棕灰色文字表示系统依据作者姓名识别的本人其他文献部分

8. 本报告单仅对您所选择的比对时间范围、资源范围内的检测结果负责



✉ amlc@cnki.net

🌐 <https://check.cnki.net/>