

摘 要

随着科学技术的发展，社会的进步，人们对生活质量要求逐渐提高，对于农业生产也提出了更高的要求。传统的农业生产模式已经无法满足现代农业发展对生产过程自动化、信息化和智能化的要求。近年来，随着社会经济技术水平和综合国力的提高和世界范围内各种信息技术迅速发展，我国提出了“走出去”战略和“一带一路”战略。随着我国经济的高速发展，现代化农业信息化、农业机械化都得到了迅速的发展和进步。为了能够及时准确地监测农田环境信息，将传感器、单片机、控制单元等设备连成网络，实现网络数据传输也是必不可少的环节。Zigbee (ZigBee 是一种基于无线网状网路、由低功耗成员机或传感器节点构成的无线网状网络) 自问世以来受到了众多研究人员的关注，目前已被广泛应用于工业控制、家庭自动化等领域。因此如何充分利用现代高科技手段实现环境信息收集与检测就显得尤为重要。

本文设计了一种基于 Zigbee 无线通信、以单片机为核心器件并结合温湿度传感器，光照传感器及二氧化碳传感器组成的农业环境感知系统；利用计算机软硬件技术实现了数据采集功能；通过 MySQL 数据库实现将数据存入数据库进行保存；pc 客户端由 c#编写，移动端由浏览器访问网页来实现，其中服务器由 node.js 进行编写，前端使用 vue 框架及各种组件来实现。本系统在实现主要功能的同时充分考虑了经济效益，达到了功率和损耗低、成本小、稳定可靠，在实际测试过程中能够稳定的实现相关功能。

关键词：农田环境监测；Zigbee；C#；MySQL；vue；node.js

ABSTRACT

With the of development of science and technology and the progress of society, people requirements for quality of life have gradually improved, and higher requirements have been put forward for agricultural production. The traditional agricultural production mode can no longer meet the requirements of modern agricultural development for the automation, informatization and intelligence of the production process. In recent years, with the improvement of socio-economic and technological level and comprehensive national strength and the rapid development of various information technologies in the world, China has put forward the "going out" strategy and the "One Belt, One Road" strategy. With the rapid development of China's economy, modern agricultural informatization and agricultural mechanization have been rapidly developed and progressed. In order to be able to monitor farmland environmental information in a timely and accurate manner, it is also indispensable to connect sensors, single-chip microcomputers, control units and other equipment into a network to realize network data transmission. Zigbee (ZigBee is a wireless mesh network based on wireless mesh networks, consisting of low-power member machines or sensor nodes) has attracted the attention of many researchers since its inception and has been widely used in industrial control, home automation and other fields. Therefore, how to make full use of modern high-tech means to achieve environmental information collection and detection is particularly important.

In this paper, an agricultural environment perception system based on Zigbee wireless communication, with a single-chip microcomputer as the core device and combined with temperature and humidity sensor, light sensor and carbon dioxide sensor, is designed. The use of computer software and hardware technology to realize the data acquisition function; Through the MySQL database, the data is stored in the database for storage; The PC client is written by C#, the mobile side is implemented by the browser to access the web page, the server is written by the node .js, and the front end is implemented using the VUE framework and various components. This system fully considers the economic benefits while realizing the main functions, and achieves low power and loss, low cost, stability and reliability, and can stably realize relevant functions in the actual test process.

Keywords: farmland environmental monitoring; Zigbee; C#; MySQL; vue; node.js

目 录

摘 要	I
ABSTRACT	II
第 1 章 绪论	1
1.1 课题的背景意义	1
1.2 国内外研究现状	2
1.2.1 国外研究现状	2
1.2.2 国内研究现状	2
1.3 论文研究目的及内容	3
1.4 论文章节安排	4
1.5 本章小结	4
第 2 章 农田环境监测系统总体设计	5
2.1 系统设计需求分析	5
2.2 系统模块分析	5
2.3 系统总体设计方案	6
2.4 本章小结	7
第 3 章 农田环境监测系统的硬件设计与实现	8
3.1 ZigBee 网络设计与实现	8
3.1.1 ZigBee 概述	8
3.1.2 ZigBee 技术特点	8
3.1.3 ZigBee 节点与拓扑结构	9
3.1.4 CC2530 芯片	12
3.2 传感器设计与实现	12
3.2.1 DHT11 温度传感器	12
3.2.2 BH1750 光照传感器	14
3.2.3 MG811 二氧化碳传感器	14
3.3 本章小结	15
第 4 章 农田环境监测系统的软件设计与实现	16
4.1 数据库设计	16
4.1.1 用户登录信息表	16
4.1.2 农田环境数据表	16

4.2	PC 端设计与实现	17
4.2.1	登录界面	17
4.2.2	实时数据展示界面	18
4.2.3	历史数据展示界面	18
4.3	移动端设计与实现	19
4.3.1	移动端服务器	19
4.3.1.1	登录接口	19
4.3.1.2	实时数据接口	20
4.3.1.3	历史数据接口	21
4.3.2	移动端客户端	23
4.3.2.1	登录模块	23
4.3.2.2	实时数据模块	24
4.3.2.3	历史数据模块	26
4.4	本章小结	27
第 5 章	农田环境监测系统测试	28
5.1	硬件测试	28
5.2	PC 客户端测试	28
5.2.1	登录功能测试	28
5.2.2	PC 端实时数据显示功能测试	29
5.2.3	PC 端历史数据显示功能测试	30
5.3	移动端功能测试	30
5.3.1	移动端服务器测试	31
5.3.1.1	登录接口测试	31
5.3.1.2	实时数据接口测试	32
5.3.1.3	历史数据接口测试	32
5.3.2	移动端客户端测试	33
5.3.2.1	登录模块测试	33
5.3.2.2	实时数据展示模块	35
5.3.2.3	历史数据展示模块	35
5.4	本章小结	36
第 6 章	总结与展望	37
6.1	总 结	37

6.2 展 望	37
参考文献.....	38

第 1 章 绪论

1.1 课题的背景意义

信息化、精准化是新时期农业发展的新要求，农业生产规模化、智能化发展，已经成为高效农业、现代化农业的一个重要组成部分^[1]。随着通信技术，网络传输技术和控制技术的高速发展全世界的农业正在从依靠劳力和农资的生产方式逐步的向低能耗、高效、信息化的新型现代化农业进行发展。在加强农业生产管理及问题决策中，如何精确、实时、高效地获取农作物在种植和生长过程中的各项重要信息是重中之重，而得到农作物生长环境信息的重要方法之一是对农田数据的采集和监测。^[2]以现代的科学技术和高效的管理理论来改造传统农业，使得传统农业的资源产出效率、土地生产效率、劳动生产率得到提高，实现新型农业的经济效益、生态效益、社会效益的统一。近几年随着信息化农业在农业生产中越来越深入，对农田中的信息获取提出了新的要求，且农田的生产环境中也存在者诸多特点，如农田作物分布范围大、周边人烟稀少、远离居住区且交通不便。所以在传统的农业生产场地常常是无人值守，农田中的环境状况感知困难，如果能对农田中的环境进行实时监测则农田管理者可根据采集的信息进行预知，为生产活动、管理决策、制定生产计划提供有效的数据。

物联网技术是一种通过信息采集设备即传感器，依靠相关的协议，将监测的物体与网络进行连接并进行数据的交换与通信，从而实现对物体的监测和管理的一种网络，该项技术对解决农田环境数据的实时采集，了解因自然环境原因导致农田减产等一系列问题都具有很重要的意义。计算机技术的发展为农业方面的智能应用提供了技术保障。当前短距离传输数据的通信技术主要有 wifi 技术、uwb 技术、IrDA 技术和 zigbee 技术等。在农田环境监测系统中需要的是一种采集数据网点多、功耗低、设备成本较低、发送的数据量小、传输可靠性高的一种传输技术，zigbee 技术就是一种短距离、低功耗、成本低的无线通讯技术，在通讯范围内可以快速通过主节点的模块通信实现数据的传输，且通过 zigbee 协议组成的传感网络，不需要大量的布线相较于其他的通讯方案来说使用该技术来实现本系统就可以实现在成本较低的同时系统的功能也能完整的实现。

在本系统中，可以通过传感器节点进行采集温度、湿度和气体浓度等数据，并且将采集到的数据通过 zigbee 终端节点发送到协议节点，并将数据发送到上位机，由数据库进行数据的存储用户可在 pc 及移动端查看相关数据，利用有限网络检测就可以实现农田温度、湿度、光照度等农作物生长环境的检测，从这些生长环境出发，从生长源头出发，以提高农作物的产量^[3]。

1.2 国内外研究现状

1.2.1 国外研究现状

近年来,越来越多的智能化系统应用于日常生活。农业智能化就是新兴的产业,如何有效提高农作物的培育效率,高效利用作物的生长期,温室环境检测与调节成为最关键的一环。^[4]对于这些问题,文献^[5-6]提到在种植农作物时,应该应用多种类的环境传感器节点对环境中的空气温湿度和光照度、二氧化碳浓度进行信息的采集与监测,将采集到的信息通过传感网络传输到远程的监测系统中。在农业传感器智能化方向,随着农业的生产越来越精准化,对农产品生长环境的要求越来越严格才能保证产品的质量^[7]所以世界范围内对农业智能化的技术发展很重视,近几年来在农业、林业、牧业的监测方面进行了大量的研究。美国完善的农业信息系统对农业生产者大有裨益^[8]美国科罗拉多大学的研究人员开发了一种智能芯片,可以嵌入植物叶子上,监测植物水分等信息,并可以将数据发送到用户的手机上。Dae-Heon Park 等人开发了温室智能监测系统,主要通过长期的数据积累,对温室中植物叶片的温度和湿度、温室中的温湿度和光照强度等进行监测,来判断植物的生长情况,并将监测数据存储在数据库中。在过去的 10 多年里,日本一直在探索智能农业的过程。用高科技发展专业化、集约化、智能化管理,建议高科学技术对农业生产的支持减少了对土壤和气候的影响等自然条件,利用先进的农业生产技术创造出农业生态模型。日本研究人员 Masayuki Hirafuji 和他的研究团队开发了一种无线方法来监测大面积的农田网络,该无线网络通过 WIFI 无线网络在传感器节点之间相互通信,收集的信息通过 GPRS 被发送到远程服务器。

1.2.2 国内研究现状

我国现代无线传感器网络及其应用研究最早于 1999 年正式发表于中国科学院的《知识创新工程试点领域方向研究》创新技术试点领域的研究在“信息与自动化研究报告”中作为该领域提出的五个要点其中一个重大项目(当时的项目名称:重点地区灾害实时监测、预警和决策支持示范系统)。2001 年中科院在上海微系统研究所建立了微系统研发中心,并陆续向无线传感器网络实施开展了多项重大研究项目。中国科学院计算技术究所在 IPv6 无线传感器网络节点和符合 IEEE802.15.4 标准和符合 ZigBee 规范的无线传感器网络上做了大量工作,自主研发的 GANIS 系列无线传感器已经推出网络节点并实现产业化。

二〇一八年中央“一号文件”的发布依然持续关注“三农”,农业作为我国的一大基础产业,是国民经济的重中之重。伴随着现代互联网和物联网以及各种高新技术的大力发展,智慧农业将成为未来农业发展的一大趋势。通过各种新兴技术的运用,提高工作效率、预防各种灾害、保障食品安全以及电子商务销售,将使得农业的发展更加高效、安全。

这对我国实现全面建设小康社会这一目标来说是一大强有力的推力^[9]在农业现代化方面我国近几年有也许多的研究案例如：

2021 年张璐璐发表在《农机化研究》上的“基于 DSP 与 ZigBee 的农田灌溉水质监测控制系统设计”

2021 年杨宇辉发表在《工业控制计算机》上的“基于 ZigBee 的农田土壤湿度监测系统”

2022 年潘坛发表在《农村使用技术》上的“基于 ZigBee 网络与嵌入式 3S 技术的农田信息监测系统设计”

1.3 论文研究目的及内容

农田环境监测系统分为两个部分，分别为可以显示数据的上位机与下位机，下位机由各个传感器与 zigbee 节点组成，各个传感器分别获取农业信息中温度、湿度、光照信息。ZigBee 节点可以作为终端节点，直接监测感知用户所要获取的目标信息，也可以作为中转节点接受其他节点发过来的信息进行数据上传。上位机的功能主要是接收下位机上传的数据并将数据存储进数据库中。pc 客户端实现各个参数的显示，在移动端可通过网站来查询，给农田管理者作为参考这样保证农田作物的生长环境能随时被系统使用者感知，能根据数据及时作出应对方案。综上所述为了完成本系统主要做了以下几点研究：

(1) 通过温湿度传感器模块、光照强度传感器模块、二氧化碳浓度传感器模块对农田的环境进行数据采集，经无线传感网络发送给协调器，然后协调器节点通过串口模块将食品贮存环境数据传输至客户端上位机系统。

(2) 使用数据库进行数据的储存，同时数据库中也存储用户的用户名及密码用于登录系统使用。

(3) 使用 C#进行 pc 客户端的编写，主要功能为从数据库中读取用户的用户名及密码实现登录功能，将获取的实时数据存入数据库中，并实现实时数据的显示。还可以查询数据库中的历史数据表，展示历史数据。

(5) 使用 node.js 编写移动端的后端服务器，为移动端提供登录接口、实时数据接口和历史数据接口。

(4) 使用 vue 框架进行移动客户端的编写，主要功能为调用后端提供的登录接口实现用户登录的功能，调用实时数据接口展示实时数据，调用历史数据接口展示历史数据。

1.4 论文章节安排

本文共分为五章，其具体的组织结构如下：

第一章 绪论。本章介绍了农田环境监测系统的背景以及研究的目的与意义，分析并阐述了目前国内外相关的专家学者对农田环境监测与应用方面的早期的相关研究与当下的研究状况，针对之前所作的研究中存在的问题进行汇总与分析，从而明确了本文的研究方向，最终表述了本文对农田环境监测系统的大体框架。

第二章 农田环境监测系统总体设计。本章主要介绍了农田环境监测系统的整体设计，根据系统的需求，初步设计其总体结构。

第三章 农田环境监测系统硬件设计与实现。本章介绍了农田环境监测系统的硬件部分的具体设计与实现，并且提供了农田环境监测系统硬件设计方案中采用的硬件配置，有利于对硬件的进一步了解。

第四章 农田环境监测系统软件设计与实现。本章介绍了农田环境监测系统软件部分的具体设计，根据需求在 pc 端与移动端的软件中实现相应的功能。并且将编写过程中的重点进行详细说明，有利于对软件编写过程的进一步了解。

第五章 农田环境监测系统测试。本章主要说明了农田环境监测系统所使用的软硬件运行时的具体情况，对软件部分与硬件部分进行了相关的测试。

第六章 总结与展望。本章主要对农田环境监测系统在测试时的不足进行了分析，并提出下一步的工作计划。

1.5 本章小结

本章表述了农田环境监测系统的背景和实际意义，并且说明了国内外相关专家学者在农田环境监测与应用上所做出的相关理论研究，对前人所做的研究中所存在的问题进行了汇总分析和总结，进而明确了本系统的设计方向，最终论述了本系统的设计思路，然后阐述了本课题论文的组织结构框架。

第2章 农田环境监测系统总体设计

2.1 系统设计需求分析

农田信息监测系统主要有三大部分组成，信息采集模块，信息传输模块，以及信息储存及展示模块分别对应物联网的感知层、网络层以及应用层。本文主要目的是希望通过物联网技术建立一套农田信息监测系统，农田信息采集及信息传输采用比较成熟的方案。信息展示模块采用比较新的技术框架。其中信息采集模块负责对农田信息的采集，比如温度、湿度、光照度、二氧化碳浓度等信息；信息传输模块可以将农田数据通过 zigbee 传输至上位机；信息储存及展示模块主要是将接收到的信息进行储存，并在 pc 客户端及移动端进行展示。

2.2 系统模块分析

（1）信息采集模块

该模块的主要功能是通过温湿度传感器、光照度传感器、二氧化碳浓度传感器实现对农田的温度、湿度、光照度、二氧化碳浓度的监测。对其的性能要求是要保证对数据的实时采集，对采集的数据精度要在要求范围之内。且成本也不宜过高，尽量采用较为简单的设计方案，这样既能实现需求还可以节省开发成本，提高其实用性和经济性。

（2）信息传输模块

该模块的主要功能是通过 zigbee 技术将信息采集模块中传感器采集到的农田环境数据传输至上位机，只有实现了数据的传输，后续的工作才能继续进行。对其性能的要求是在进行农田环境传输的过程中要保证稳定可靠。如经常出现故障，数据的传输会无法正常进行，除此之外功耗也要尽可能的低，因为该系统主要是用于农田环境的监测较低功耗也可以延长使用的寿命。在实现上述要求的同时成本也不宜过高，以提高整个系统的实用性与经济性。

（3）信息储存及展示模块

该模块分为两个部分，分别为信息储存及信息展示。其中信息储存模块主要的功能是将上位机接收的数据进行储存。对其性能要求是功能要强大，能为多种语言提供 API，服务要稳定。在实现上述需求是该数据库最好是免费数据库，这

样以节省开发成本。信息展示模块主要的功能是提供登录、实时数据展示及查询历史数据的功能。该模块还分为 pc 端及移动端，且数据互通，能使得用户可以随时随地的查看数据。

系统功能简图如图 2-1 所示

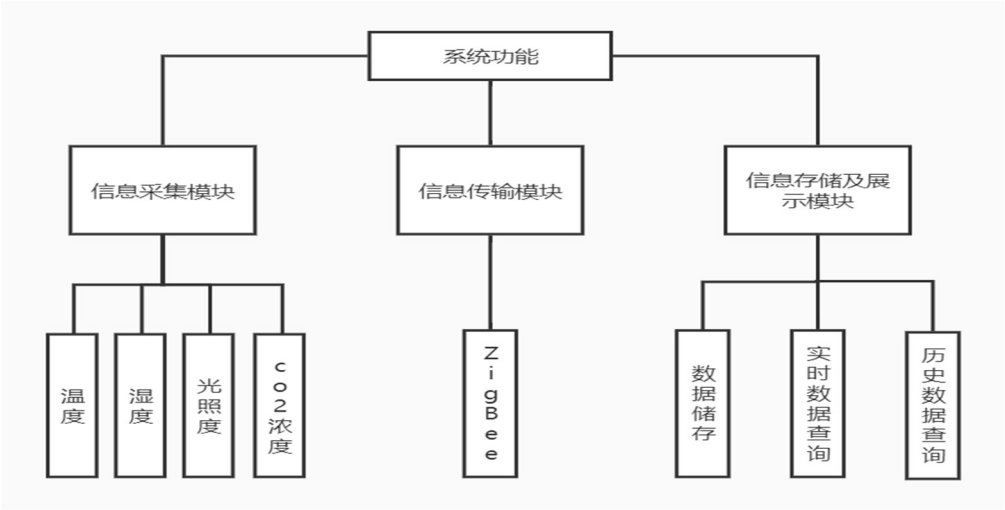


图 2-1 农田环境监测系统功能

2.3 系统总体设计方案

本文所述的农田环境监测系统主要是对农田中洋相作物生长的环境因素进行实时的采集与监测，根据上面的章节分析该系统有三层构成，分别为信息采集模块、信息传输模块和信息储存及展示模块，分别对应物联网的感知层、网络层和应用层。系统的整体框架图如下图 2-2 所示。

(1) 感知层。信息采集模块作为系统的最底层，主要是借助各种传感器来采集农田环境的各项数据。

(2) 传输层。信息传输模块由 ZigBee 组成，主要的功能是负责数据的传输，把来自信息采集模块的数据上传至上位机。

(3) 应用层。信息储存及展示模块主要分为两个个部分。分别是负责数据储存的数据库和负责数据展示的客户端



图 2-2 系统整体框架

2.4 本章小结

本章对农田环境监测系统进行了整体设计。首先对农田环境监测系统所需的各个模块进行了整体的分析。明确了每个模块的功能并对每个模块的性能及其他属性作出了要求为下文硬件及软件的选择与设计提供了参考。此外对该系统的体系结构进行初步设计。

第 3 章 农田环境监测系统的硬件设计与实现

3.1 ZigBee 网络设计与实现

3.1.1 ZigBee 概述

ZigBee 技术是 ZigBee 联盟制定的一种无线通信标准，ZigBee 联盟指定的 ZigBee 标准包括 4 层：物理层、MAC 层、网络层和应用层。其中物理层是 IEEE802.15.4 工作组制定的，ZigBee 联盟只定义了其网络层与应用层。^[10] ZigBee 网络结构如图 3-1 所示

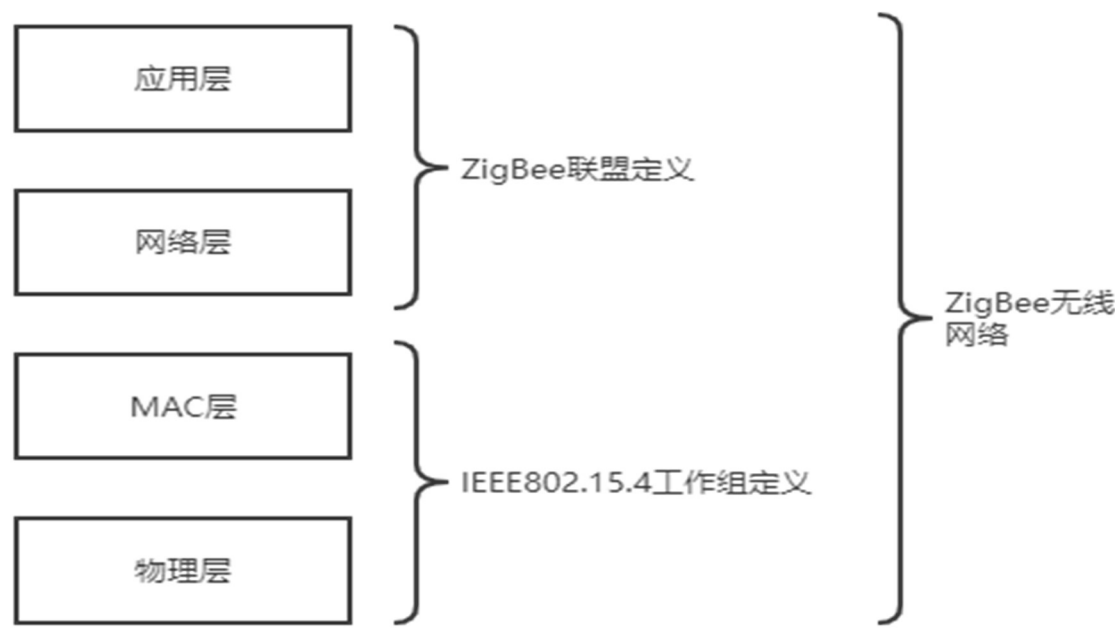


图 3-1 ZigBee 网络结构

3.1.2 ZigBee 技术特点

随着通信技术的快速发展，通信技术种类越来越多。在实际应用中常见的通信技术有 ZigBee、Wi-Fi、蓝牙、和 RFID 技术。基于对现有技术的发展和补全，ZigBee 技术也由此诞生，多以与其他的通讯技术有一定的区别，下表是对这几种常用的通信技术的比较，如表 3-1 所示

表 3-1 常见的通信技术性能比较

规范	Zigbee	Wi-Fi	蓝牙	RFID
最大功耗	1~3mW	100mW	1~100mW	接口
传输速率	250kb/s	300Mb/s	3Mb/s	200Kb/s
传输距离	10~100m	100~300m	10m	低于 10m
安全等级	中	低	高	中
频段	2.4GHz	2.4GHz,5GHz	2.4GHz	5.8GHz

ZigBee 技术能被应用在农田环境监测系统中是因为有以下几种优点

- (1) 自组网: ZigBee 网络中的设备在现场配置就能加入网络, 方便安装。
- (2) 低成本: ZigBee 模块的初始成本在 100 元左右, 而且 ZigBee 技术是免专利费的, 代码简单大多数开源, 可以免费获取。这样可以降低系统的整体开发成本
- (3) 低功耗: 由于 ZigBee 的传输速率低, 发射功率仅有 1mW, 而且还采用了休眠模式, 因此 ZigBee 设备很省电。供电方面仅靠两节五号电池就可以保障设备长时间工作
- (4) 可靠: 采用了碰撞避免策略, 同时为需要的固定带宽的通信业务预留了时隙, 避开了发送数据的冲突。MAC 层采用了完全确认的数据传输模式, 每个发出的数据包都必须等待对方确认接收消息, 传输过程中出现错误还可重新发送。

3.1.3 ZigBee 节点与拓扑结构

ZigBee 共有三种类型的节点, 分别为 ZigBee 终端节点, ZigBee 路由节点和 ZigBee 协调器节点, 以下是对这三种节点的详细介绍:

(1) 协调器节点

协调器节点是 ZigBee 网络中最重要节点, 通过协调器节点可以完成网络的创建和初始化功能。协调器在工作时会初始化整个网络并扫描所有的信道情况从中挑选一个合适的信道用于创建网络, 并且将路由节点传来的数据发送至上位机。是整个网络中的主控节点

(2) 路由节点

路由节点在主要负责数据的传递功能, 在网络中起到中继的作用。路由节点可以接收终端节点传来的数据, 将接收的数据上传至协调器节点。

(3) 终端节点

终端节点主要是负责接收传感器采集的数据, 并将数据发送给路由节点或传

传感器节点，终端节点不能创建网络。

ZigBee 的拓扑结构一共有三种分别是星状拓扑结构，树状拓扑结构，网状拓扑结构^[11]以下是这三种拓扑结构的介绍：

(1) 星状拓扑结构

形状拓扑结构由终端节点与协调器节点构成，只能多个终端节点对一个协调器节点进行数据传输，整体结构比较简单节点布置与网络管理较为方便。该拓扑结构的拓扑图如图 3-2 所示。

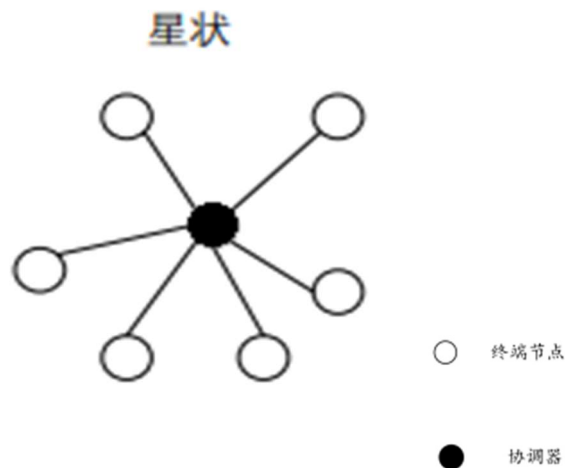


图 3-2 星状拓扑结构

(2) 树状拓扑结构

包括了终端节点、路由节点和协调器节点组成，协调器是网络的第一层，其他的节点都拥有唯一的根节点，每个根节点可以存在较多的子节点，数据可以按照分组进行传输，每组数据间不会相互影响。其拓扑图如下图 3-3 所示。

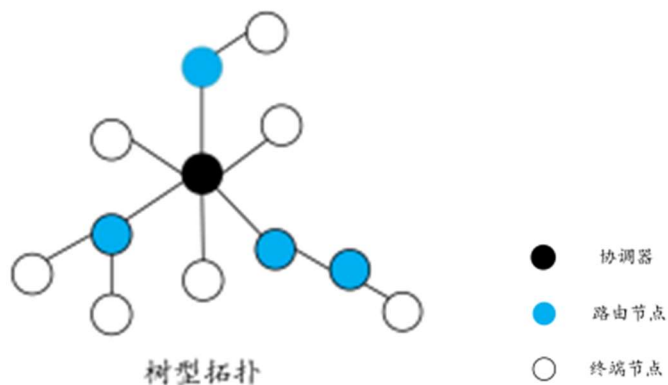


图 3-3 树状拓扑结构

(3) 网状拓扑结构

该拓扑结构也是由终端节点、路由节点、以及协调节点组成。在该结构中任意一点发出广播数据后，网络中的其他点都能收到，并可以自主构建路由和维护网络。该拓扑结构的拓扑图如图 3-4 所示。

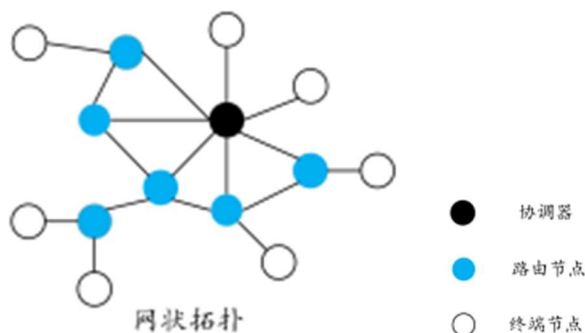


图 3-3 树状拓扑结构

本文使用了星状的拓扑结构，其中终端节点向协调器节点发送数据的核心代码如下：

```
/*禁止 RF 中断*/
IEN2 &=~(1<0);
/*清除发送区缓存*/ REST =0xee;
/*清除 TXDONE 中断 */ RFIRQF1=~(1<<1);
// 延时 1S
for(i=20; i>0;i-- )
delay_nus(50000);
//得出温度、湿度数据
SendRH=RH();
//获取光照度数据
Light=read_BH1750();
//获取二氧化碳数据
CO2=read_MG811();
DATA[0]=" FF" ;
DATA[1] =sendRH[0];
DATA[2] =sendRH[1];
DATA[3] =Light;
DATA[4] =CO2;
/*传输的帧长度*/
RFD = 10;
/*将 TEMP 的内容写到 RFD 中*/
for(i=0;i<4;i++){
RFD = DATA[i];
}
```

由协调器接收终端节点数据通过串口向上位机发送数据的核心代码如下：

```
(RFIRQFO & (1<<6)) {
len = RFD;
len = 0x7f;
```



```

for(i = 0;i<len;i++){
    buf[i] = RFD;
    Delay(200);
}
//清 RF 中断
S1CON = 0;
//清除 RXPKTDONE 中断
RFIRQFO &= ~(1<<6);
//调用串口发送函数
Send(buf, len)
}

```

3.1.4 CC2530 芯片

本系统采用了 CC2530_F256 作为 ZigBee 的控制芯片，该芯片具有多种外设，提供了完善的 ZigBee 解决方案，芯片电路图如图 3-4 所示

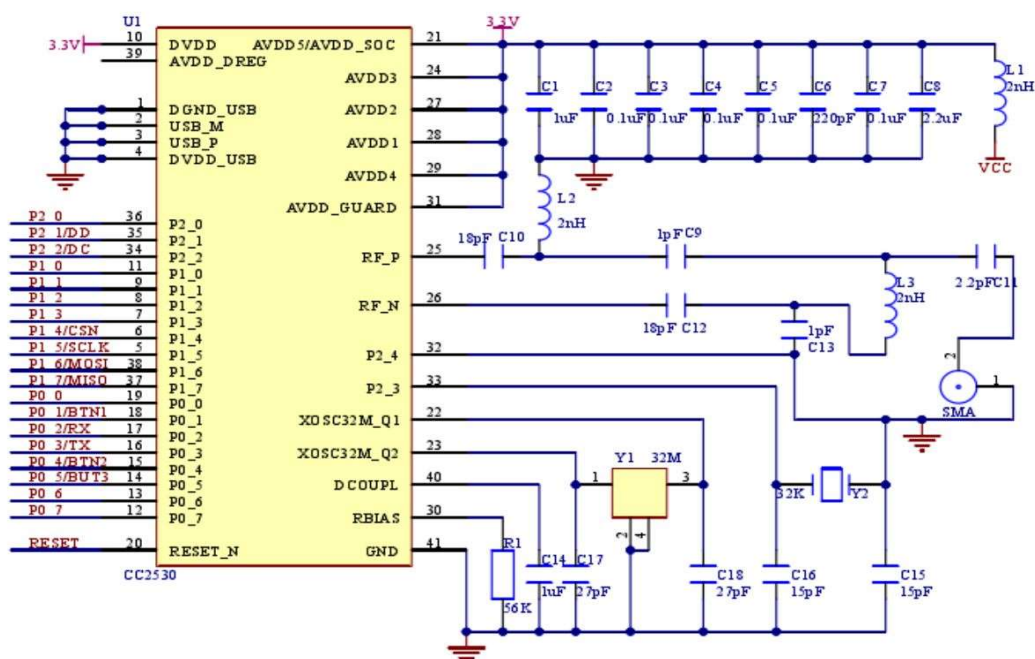


图 3-4 CC2530_F256 电路原理图

3.2 传感器设计与实现

3.2.1 DHT11 温度传感器

DHT11 数字湿度和温度传感器是具有校准数字信号输出的复合温湿度传感

器，它采用特殊的数字模块采集技术和温湿度测量技术，确保高可靠性和出色的长期稳定性^[12]。该传感器实物图以及电路图如图 3-5，3-6 所示。

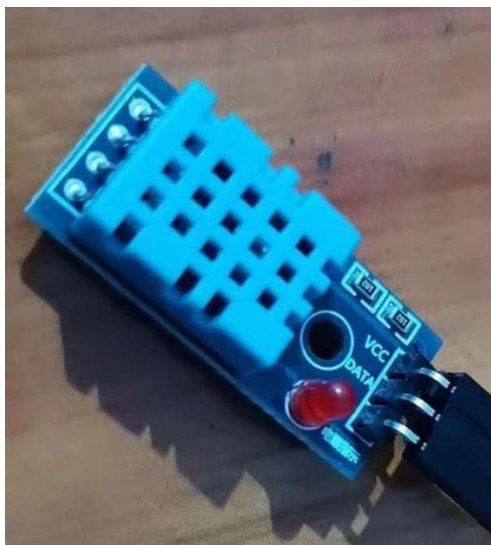


图 3-5 DHT11 温湿度传感器

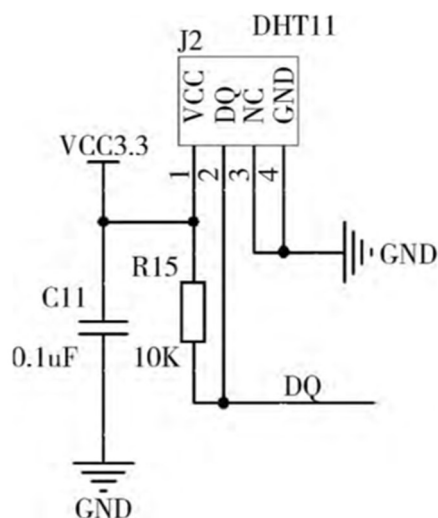


图 3-6 DHT11 电路图

DHT11 温度传感器获取农田环境温湿度的主要代码为：

```
//主机设为输入 判断从机响应信号
P2_0=1;
//判断从机是否有低电平响应信号
if(!P2_0){
    U8FLAG=2;
    //判断从机是否发出 80us 的低电平响应信号是否结束
    while((!P2_0)&&U8FLAG++);
    U8FLAG=2;
    //判断从机是否发出 80us 的高电平，如发出则进入数据接收状态
    while((P2_0)&&U8FLAG++);
    //数据接收状态
    COM();
    U8RH_data_H_temp=U8comdata;
    COM();
    U8T_data_H_temp=U8comdata;
    COM();
    U8checkdata_temp=U8comdata;
    P2_0=1;
    if(U8temp==U8checkdata_temp)
    {
        U8RH_data_H=U8RH_data_H_temp;
        U8T_data_H=U8T_data_H_temp
    }
    GetRH[0]= U8RH_data_H
    GetRH[1]= U8T_data_H
```

```
return GetRH;
```

3.2.2 BH1750 光照传感器

BH1750 是一种用于两线式串行总线接口（IIC）的数字型光强度传感器集成电路，这种集成电路可以检测光线强度，其中有 16 位模数转换器，将其转化为数字信号。通过计算电压来获得有效的数据。这款环境光传感器能够直接通过光度计来测量^[15]。该传感器实物图以及电路图如图 3-7，3-8 所示

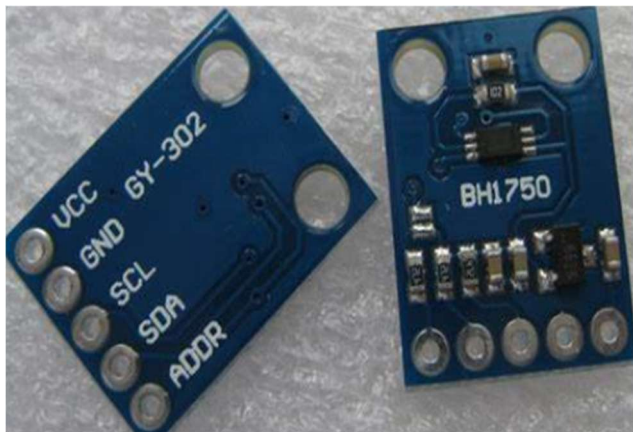


图 3-7 BH1750 光照传感器

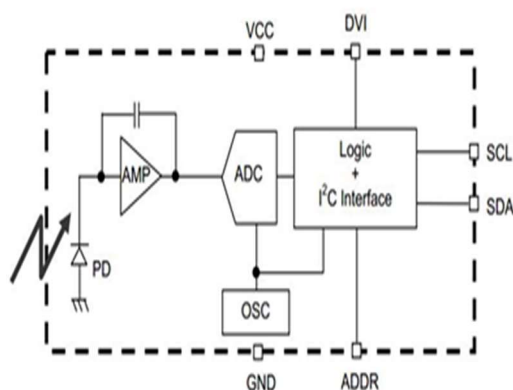


图 3-7 BH1750 电路图

BH1750 传感器获取农田环境中光照度的主要代码为：

```
int g_data;           //变量
float temp;
Single_Write_BH1750(0x01); //发送上电命令(0x01)
delay_ms(200);
mread();             //连续读出数据，存储在 BUF 中
g_data=BUF[0];
g_data=(dis_data<<8)+BUF[1]; //2 个字节合成数据
temp=g_data/1.2; //计算光照度
return temp;
```

3.2.3 MG811 二氧化碳传感器

MG811 电解质 CO₂ 传感器具有对 CO₂ 有良好的灵敏度和选择性，通过测试信号电压的变化来检测 CO₂ 浓度的变化受温湿度的变化影响较小，良好的稳定性、再现性等特点^[16]。该传感器实物图以及电路图如图 3-9，3-10 所示。

MG811 在农田中获取农田环境中获取二氧化碳浓度的主要代码为

//通过计算电压来获得 CO₂ 浓度值

```
float c_data= 0;
c_data = (_V400 - _V40000)/(log10(400) - log10(40000));
c_data = (raw() - _V400)/buffer;
c_data += log10(400);
```

```
return pow(10, c_data);
```

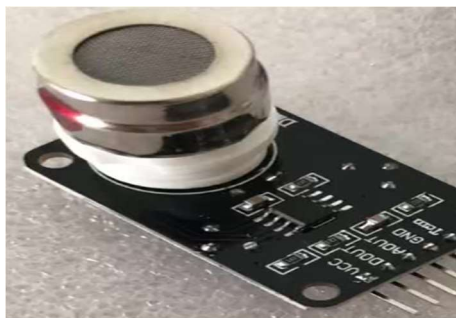


图 3-9 MG811 光照传感器

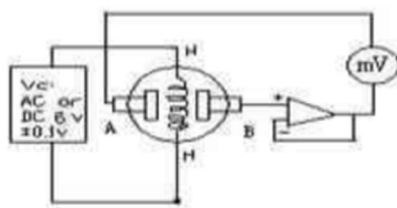


图 3-10 MG811 电路图

3.3 本章小结

本章介绍了农田环境监测系统的硬件部分的设计与实现,详细介绍了 ZigBee 的技术特点,以及农田环境监测系统所使用的传感器的特点,并且给出了传感器的实物图,与电路原理图及其实现功能的核心代码,有利于进一步了解农田环境监测系统的硬件设计。

第 4 章 农田环境监测系统的软件设计与实现

4.1 数据库设计

农田环境监测系统使用的是 MySQL 数据库。该数据库是开源数据库，使用该数据库可以有效降低开发成本，除此之外 MySQL 数据库还具备体积小、速度快的优点^[17]。综上所述 MySQL 数据库满足系统设计需求，所以采用 MySQL 数据库作为本系统的数据库。

4.1.1 用户登录信息表

用户信息表主要涉及 id、用户名、密码两种信息，如图 4-1 所示。

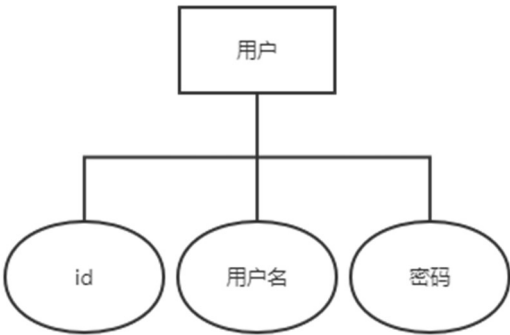


图 4-1 用户信息 E-R 图

建立用户登录信息表 `ev_users`，具体表结构如图 4-2 所示

	字段	类型	整理	属性	空	默认	额外	操作
<input type="checkbox"/>	id	int(11)			否	无	AUTO_INCREMENT	     
<input type="checkbox"/>	username	varchar(255)	gb2312_chinese_ci		否	无		     
<input type="checkbox"/>	password	varchar(255)	gb2312_chinese_ci		否	无		     

图 4-2 `ev_users` 表

4.1.2 农田环境数据表

农田环境数据表存储了下位机上传的农田环境数据，该表主要涉及 id、温度、湿度、光 CO2 浓度和时间六种信息，如图 4-3 所示

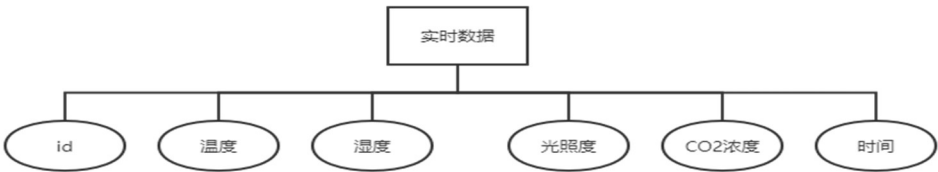


图 4-3 农田环境信息 E-R 图

建立农田环境数据表 f_data, 具体表结构如图 4-4 所示。

	字段	类型	整理	属性	空	默认	额外	操作						
<input type="checkbox"/>	id	int(11)			否	无	AUTO_INCREMENT							
<input type="checkbox"/>	wendu	int(3)			否	无								
<input type="checkbox"/>	shidu	int(3)			否	无								
<input type="checkbox"/>	guangzhaodu	int(6)			否	无								
<input type="checkbox"/>	co2	int(3)			否	无								
<input type="checkbox"/>	Date	date			否	无								

图 4-4 f_data 表

4.2 PC 端设计与实现

PC 端使用 C#进行编写, 相较其他语言 C#有更先进的语言体系, 提供了大量的功能与接入, 让开发更高效。开发环境为 Microsoft Visual C# 2010 Express

4.2.1 登录界面

PC 端登录界面主要负责用户的登录, 实现的步骤如下:

(1) 用户输入用户名与密码, 系统使用用户名与密码作为数据库查找条件, 将查找的结果存入表单中, 判断表单中是否有数据。如果有数据则表示用户名与密码正确可以跳转至其数据接收界面, 如果没有数据则表示用户名与密码有错误, 系统会提示“用户名或密码错误”的提示框

(2) 验证码, 系统将在验证码生成区域随机生成由字母、数字和噪音点、线组成的验证码, 如果用户输入的验证码不正确, 系统会提示“验证码错误”。

(3) 只有用户输入正确的用户名、密码和验证码时系统才会判定登录成功, 有一项输入错误都不会跳转至其他界面。

PC 端实现登录功能的流程图如图 4-5 所示。

登录功能实现的核心代码为:

//连接数据库

```
MySqlConnection conn = new MySqlConnection("data source =127.0.0.1;database=iot;user id=root;password=123456");
```

```
conn.Open();
```

//以输入的用户名密码作为查找条件在 ev_user 表中进行查找

```
MySqlDataAdapter da = new MySqlDataAdapter("select * from ev_users where username = '" + textBox1.Text + "' and password = '" + textBox2.Text + "'", conn);
```

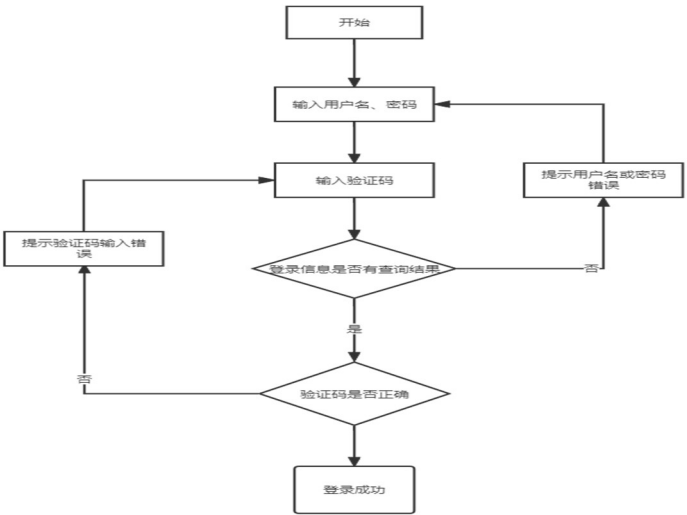


图 4-5 登录功能流程图

4.2.2 实时数据展示界面

实时数据界面主要负责的功能是将接收到的农田环境数据进行实时的展示，实现的步骤如下：

- (1) 接收来自下位机的数据。
- (2) 将下位机传来的数据进行处理，完成处理后进行展示。
- (3) 将数据存入数据库。

实现功能的具体代码如下图 4-6 所示。

```
private void AddContent(string content)
{
    this.BeginInvoke(new MethodInvoker(delegate
    {
        textBox1.AppendText(content);
    }));
}

private void serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    byte[] ReData = new byte[serialPort1.BytesToRead];
    serialPort1.Read(ReData, 0, ReData.Length); //读取数据
    string rsStr = Encoding.ASCII.GetString(ReData);
    AddContent(rsStr);
    DateTime time = new DateTime();
    time = DateTime.Now;
    textBox4.Text = time.ToString();
    conn.Open();
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    if (textBox1.Text.Substring(0, 2) == "FF")
    {
        textBox2.Text = textBox1.Text.Substring(2, 2);
        textBox3.Text = textBox1.Text.Substring(4, 2);
        textBox5.Text = textBox1.Text.Substring(6, 3);
        textBox6.Text = textBox1.Text.Substring(9, 3);
        MySqlCommand cmd = new MySqlCommand("insert into f_data values ('", "'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + textBox4.Text + "')");
        cmd.ExecuteNonQuery();
        //清空数据接收位
        textBox1.Text = "";
    }
    else {
        MessageBox.Show("数据接收错误");
    }
}
```

图 4-6 PC 端实时数据展示代码

4.2.3 历史数据展示界面

历史数据展示界面负责的功能就是展示之前的农田环境数据，实现的步骤如

下:

- (1) 将数据从 `f_data` 表中读出。
- (2) 将读出的数据放入表单中。
- (3) 在界面的规定范围内进行展示。

实现过程的核心代码为:

```
MySqlDataAdapter mda = new MySqlDataAdapter("select * from f_data", conn);  
DataTable dt = new DataTable();  
mda.Fill(dt);  
dataGridView1.DataSource = dt;
```

4.3 移动端设计与实现

农田环境监测系统的移动端是由移动设备访问网页来实现的,因此将分为网页的后端与前端来分别介绍。

4.3.1 移动端服务器

移动端的服务器使用 `Node.js` 进行编写,优点主要是开发周期短,此外 `Node.js` 的语法完全是 `js` 语法,因此学习成本也低^[18]。开发环境为 `HBuilderX`

4.3.1.1 登录接口

登录接口主要的功能是给客户端提供登录服务,且生成 `token` 字符串。其中 `token` 字符串主要是用来进行权限验证。主要的实现步骤为:

- (1) 客户端通过 `post` 请求将用户名与密码发往服务端。
- (2) 服务端接收用户名、密码,验证表单数据后使用用户名在 `ev_users` 中进行查询。
- (3) 如果没有查询到结果,通过 `res.send()` 向客户端返回失败代码“1”,以及“该用户未注册”的提示。如果查询到结果则进行密码比对如果密码比对错误也要向客户端返回失败代码“1”以及“密码输入错误的提示”。如果都正确则向客户端返回登录成功的代码“0”以及“登录成功的提示”和 `token` 字符串。

登录接口实现功能的流程图如图 4-7 所示。

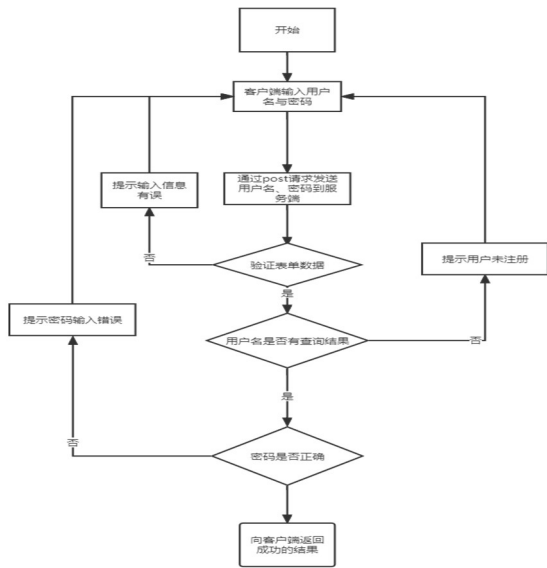


图 4-7 登录服务流程图

实现登录服务功能的核心代码为：

```
//todo 判断密码是否正确
const compareResult =
bcryptjs.compareSync(userinfo.password,results[0].password)
if(!compareResult){
    return res.cc('密码错误')
}
//在服务器端生成 token 字符串
const user = {...results[0],password:'',user_pic:''}
//对用户的信息进行加密，生成 token 字符串
const tokenstr = jwt.sign(user,config.jwtSecretKey,{expiresIn:config.expiresIn});
//调用 res. send 将 token 响应给客户端
res.send({
    status:0,
    message:'登陆成功',
    token:'Bearer '+tokenstr
})
```

4. 3. 1. 2 实时数据接口

实时数据接口主要的功能是向客户端提供查询实时数据的服务，主要实现的功能是当用户登录获取权限后向该接口发起 get 请求时，服务器向客户端发送最新的数据。若没有登录获取权限则不能调用实时数据接口获取数据。实现该功能的步骤为：

- (1) 登录，获取 token 字符串
- (2) 发起 get 请求，将 token 字符串发送给服务器，获取接口调用权限
- (3) 服务器查询最新的数据，查询成功返回成功代码“0”并且把数据响应

给客户端，若查询失败则返回失败代码“1”并且响应给客户端失败原因。

实时数据接口实现功能的步骤如图 4-8 所示。

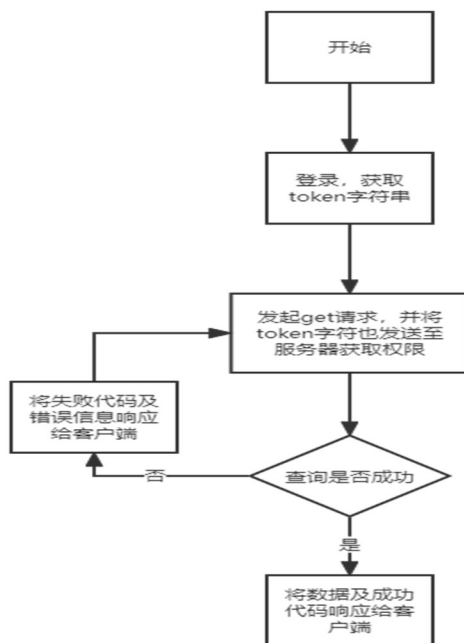


图 4-8 实时数据查询服务流程图

实现实时数据查询服务的主要代码为：

```

const sqlstr = 'select * from f_data order by id desc limit 1'
//使用 db.query 执行 sql 语句
db.query(sqlstr, (err, results) => {
  //sql 语句执行失败
  if (err) {
    return res.cc(err);
  }
  //执行 sql 语句成功
  res.send({
    status: 0,
    message: '获取成功',
    data: results
  })
})

```

4.3.1.3 历史数据接口

历史数据接口的主要功能是向客户端提供查询历史数据的服务，实现的功能的步骤为：

- (1) 登录后，获取 token 字符串。
- (2) 发起 get 请求，将 token 字符串一并发送给服务器，获取接口调用权限。

(3)服务器收到请求后查询历史记录,查询成功向客户端响应成功代码“0”并且将查询数据一并响应给客户端。查询失败向客户端响应失败代码“1”并且将错误信息响应给客户端。

历史数据接口实现功能的步骤如图 4-9 所示。

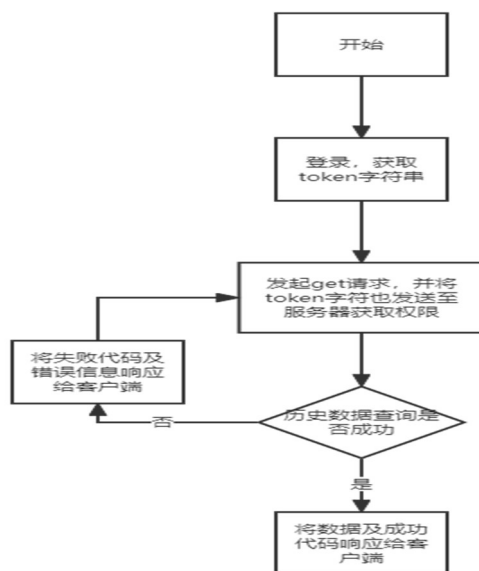


图 4-9 历史数据查询服务流程图

实现历史数据查询服务的主要代码为:

```

exports.getarticleCates = (req, res) => {
  //定义 sql 语句
  const sqlstr = `select * from f_data order by id asc`
  //调用 db.query 执行 sql 语句
  db.query(sqlstr, (err, results) => {
    //执行 sql 语句失败
    if (err) {
      return res.cc(err);
    }
    //执行 sql 语句成功
    res.send({
      stetus: 0,
      message: '获取环境数据列表成功',
      data: results
    })
  })
})

```

4.3.2 移动端客户端

移动端的使用了 vue 框架, Vue.js 是一款简单而功能强大的 JavaScript 库, 是一款面向 Web 前端的开源的框架, 能够实现软件工程项目的高效开发^[19]。开发环境为 HBuilderX。

农田环境监测系统的移动端是单页面应用, 单页面应用程序 (SPA) 是 1 加载单个 HTML 页面并在用户与应用程序交互时动态更新该页面的 Web 应用程序。浏览器一开始会加载必须的 HTML、CSS 和 JavaScript, 所有的操作都在一张页面上完成, 由 JavaScript 来控制。因此, 对单页面应用来说模块化和设计显示相当重要。

移动客户端一共分为三个模块, 分别是登录模块、实时数据模块和历史数据模块。使用 Vue-router 来控制显示的模块, Vue-router 是 Vue 的核心组件, 主要是作为 Vue 的路由管理器, Vue-router 默认 hash 模式, 即通过使用 URL 的 hash 来模拟一个完整的 URL, 当 URL 改变页面时不会重新加载。只会将对应的模块在路由占位符<router-view>进行显示。

4.3.2.1 登录模块

客户端登录模块的主要功能是获取用户的用户名、密码, 将用户名、密码通过校验后发往后端服务器, 如果返回的是失败代码 (1) 则表示登录失败, 提示用户登录失败的原因。如果返回的是成功代码 (0) 则表示登录成功, 将服务器响应来的 token 字符串进行保存, 并且跳转至实时数据界面。

在登录模块编写时遇到了__ob__: Observer 这个属性带来的无法获取表单数据的问题, 经查询得知__ob__: Observer 是 vue 这个框架对数据设置的监控器, 一般是不可枚举的。本文对于该问题使用的解决方案是使用 URLSearchParams, 该接口定义一些实用的方法来处理 URL 的查询字符串, URLSearchParams() 返回一个 URLSearchParams 对象。该接口不继承任何属性。具体的解决代码为:

```
//解决 Vue __ob__: Observer 问题
let param = new URLSearchParams()
param.append('username', this.form.username)
param.append('password', this.form.password)
```

登录功能具体的步骤如下:

- (1) 用户输入用户名、密码。
- (2) 通过校验后调用登录的 API 接口将用户名、密码发往服务器。
- (3) 接收服务器响应的数据。
- (4) 判断是否登录成功, 若登录成功则将服务器响应的 token 字符串保存

跳，并转至实时数据显示界面。若登录失败提示失败的原因。

登录功能流程图如图 4-10 所示。

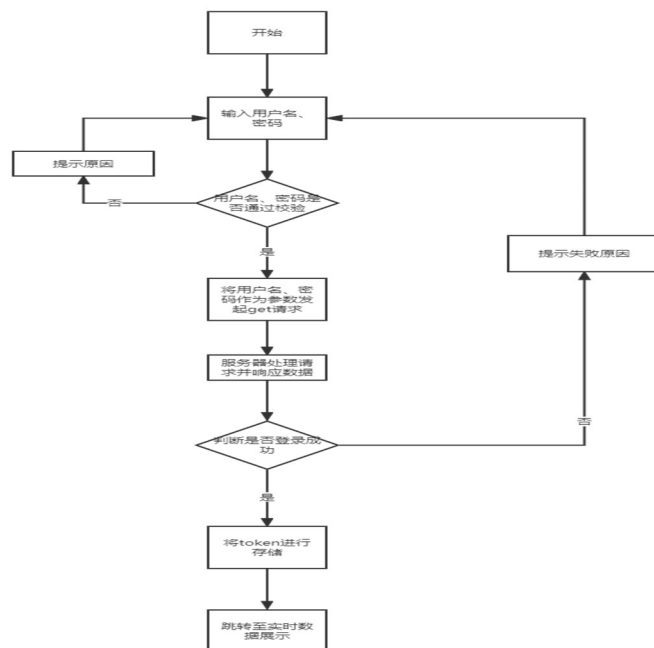


图 4-10 登录功能流程图

登录功能主要的代码如下：

```

let param = new URLSearchParams()
param.append('username', this.form.username)
param.append('password', this.form.password)
//使用 getLoginAPI() 方法发起登录请求
const {
  data: res
} = await getLoginAPI(param)
//打印返回结果
console.log(res);
//TODO:判断如果登录成功将 token 存入 sessionStorage 中，并且跳转到/user 中
if (res.status === 0) {
  window.sessionStorage.setItem("token", res.token)
  this.$router.push('/User')
} else {
  alert(res.message)
}
  
```

4.3.2.2 实时数据模块

实时数据模块的主要功能是调用实时数据接口发起 get 请求，将 token 字符串作为参数发往服务器获取权限。将服务器返回的数据进行判断，如果返回的代码为（1）代表查询失败，并展示失败原因。如果返回的代码为“0”代表展示成

功，并将响应的数据进行展示。实现上述功能的步骤如下：

- (1) 调用实时数据接口发起 get 请求，将 token 字符串作为参数发往服务器。
- (2) 服务器响应客户端请求。
- (3) 判断服务器响应的结果若为成功代码“0”则展示数据，若是失败代码“1”则提示错误信息。

步骤图如图 4-11 所示。

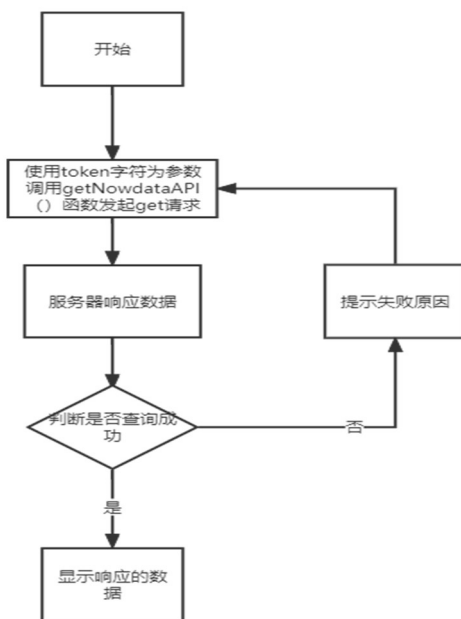


图 4-11 实时数据展示功能流程图

其核心代码如下所示：

```

const token = window.sessionStorage.getItem("token")
//console.log(token);
//调用 getNowdataAPI 发起 get 请求，传参数 token 获取权限
const {
  data: res
} = await getNowdataAPI(token)
if (res.status === 0) {
  //解决 Observer 型数据无法读取
  let getdata = JSON.parse(JSON.stringify(res.data))
  console.log(getdata);
  //将时间字符串传给 date
  this.date = getdata[0].Date
  this.list = getdata
} else {
  alert(res.message)
}
  
```

4.3.2.3 历史数据模块

历史数据模块的主要功能是调用历史数据接口发起 get 请求，将 token 字符串作为参数发往服务器获取权限。将服务器返回的数据进行判断，如果返回的代码为“1”代表查询失败，并展示失败原因。如果返回的代码为“0”代表展示成功，并将响应的数据进行展示。实现上述功能的步骤如下：

(1) 调用历史数据接口发起 get 请求，将 token 字符串作为参数发往服务器。

(2) 服务器响应客户端请求。

(3) 判断服务器响应的结果若为成功代码“0”则展示数据，若是失败代码“1”则提示错误信息。

步骤图如图 4-12 所示。

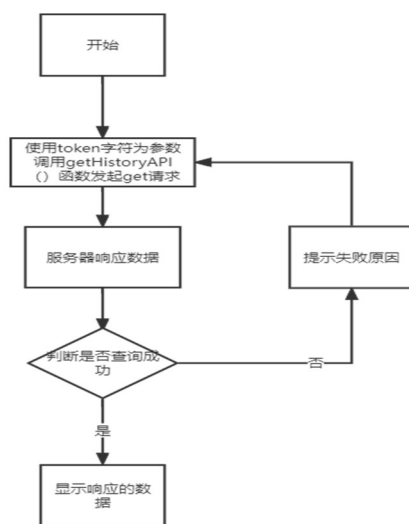


图 4-11 实时数据展示功能流程图

历史数据模块的核心代码为：

```

//获取存在 sessionStorage 中的 token, 从而获取权限
const token = window.sessionStorage.getItem("token")
//console.log(token);
//调用 getHistoryAPI() 发起 get 请求，传参数 token 获取权限
if (res.status === 0) {
  const {
    data: res
  } = await getHistoryAPI(token)
  this.list = res.data
}
} else {
  alert(res.message)
}

```

以上为客户端接收历史数据，历史数据的展示功能是使用自定义组件 `DataInfo` 进行展示。具体步骤为：

(1) 在 `History.vue` 中导入并注册 `DataInfo` 组件。

(2) 使用 `v-for` 遍历历史数据，通过自定义属性的方法将每条数据中的温、度、湿度、光照度、`CO2` 浓度数据发送给 `DataInfo` 组件。

(3) `DataInfo` 组件接收数据进行数据展示

核心代码有以下几条

导入 `DataInfo` 组件：`import DataInfo from '@components/Article/DataInfo.vue'`

注册 `DataInfo` 组件：`components: {`

`//注册组件`

`DataInfo`

`}`

将历史数据发送给 `DataInfo` 组件：`<DataInfo v-for="item in`

`list" :key="item.id" :wendu="item.wendu" :shidu="item.shidu":guangzhaodu="item.guangzhaodu" :eryanghuatan="item.co2" :pubdate="item.Date.substring(0,10)">`

`</DataInfo>`

4.4 本章小结

本章主要对农田环境监测系统的软件部分进行详细的介绍，分别介绍了系统的数据库结构、PC 端的设计与实现过程、移动端服务器和移动端客户端的设计与实现过程，同时也给出了每个功能的主要代码。通过本章的素数更有助于更加深入了解农田环境监测系统的软件的设计与实现过程。

第 5 章 农田环境监测系统测试

通过上文的叙述，我们已经实现了农田环境监测系统的搭建，本章主要对农田环境监测系统的各项功能进行测试，以检验系统能否正常运行。

5.1 硬件测试

本小节主要测是硬件系统能否正常的传递数据，测试工具为串口调试助手。测试结果如图 5-1 所示。



图 5-1 硬件数据传输结果图

通过测试串口调试助手可以接收到数据，硬件系统工作正常。

5.2 PC 客户端测试

本小节主要对 PC 客户端的功能进行测试主要测试功能为登录功能、实时数据展示功能、历史数据展示功能。

5.2.1 登录功能测试

登录功能主要测试有以下几点：

- (1) 输入错误的密码进行测试，测试结果如图 5-2 所示。

- (2) 输入错误额验证码进行测试，测试结果如图 5-3 所示。
- (3) 输入正确的密码和验证码进行测试，测试结果如图 5-4 所示。

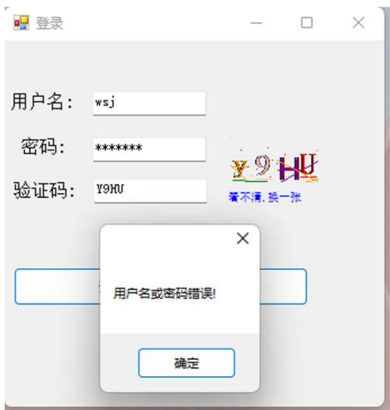


图 5-2 测试 1



图 5-3 测试 2



图 5-4 测试 3

以上测试均可以正常显示通知信息，说明 PC 端登录功能运行正常。

5. 2. 2 PC 端实时数据显示功能测试

实时数据展示主要测试为以下两点：

- (1) 能否正常接收数据，测试结果如图 5-5 所示。
- (2) 能否将数据写入数据库中，测试结果如图 5-6 所示。

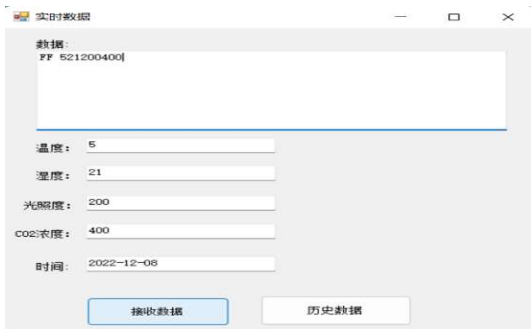


图 5-5 测试 1

20	4	50	200	400	2022-11-19
21	0	53	200	400	2022-11-20
22	0	53	200	401	2022-11-21
23	0	53	200	401	2022-11-22
24	0	52	200	401	2022-11-23
25	0	52	200	401	2022-11-24
26	0	52	200	401	2022-11-25
27	0	52	200	401	2022-11-26
28	0	52	200	400	2022-11-27
29	-1	52	200	400	2022-11-28
30	-1	53	200	400	2022-11-29
31	3	52	200	400	2022-11-30
32	3	50	200	399	2022-12-01
33	3	50	200	399	2022-12-02
34	4	51	200	400	2022-12-03
35	4	51	200	400	2022-12-04
36	5	51	200	400	2022-12-05
37	5	51	200	400	2022-12-06
38	5	21	200	400	2022-12-07
39	5	21	200	400	2022-12-08

图 5-6 测试 2

以上测试数据显示正常，数据也能写入进数据库，说明 PC 端实时数据显示功能运行正常。

5.2.3 PC 端历史数据显示功能测试

历史数据显示主要是将查询到的历史数据进行展示，主要测试的就是数据能否正常显示，其测试结果如图 5-7 所示。

id	wendu	shidu	guanghaodu	co2	Date
19	4	50	200	400	2022/11/18
20	4	50	200	400	2022/11/19
21	0	53	200	400	2022/11/20
22	0	53	200	401	2022/11/21
23	0	53	200	401	2022/11/22
24	0	52	200	401	2022/11/23
25	0	52	200	401	2022/11/24
26	0	52	200	401	2022/11/25
27	0	52	200	401	2022/11/26
28	0	52	200	400	2022/11/27
29	-1	52	200	400	2022/11/28
30	-1	53	200	400	2022/11/29
31	3	52	200	400	2022/11/30

图 5-7 历史数据查询测试

以上历史数据显示正常，说明 PC 端历史数据显示功能运行正常。

通过对农田环境监测系统的 PC 端进行测试后，测试均可正常显示数据，据此可得出结论，农田环境监测系统的 PC 端可以正常运行。

5.3 移动端功能测试

移动端的功能测试主要分为两点，分别对服务器所提供的接口和客户端进行测试，查看各项功能是否能够按照设计正常运行。

5.3.1 移动端服务器测试

对服务端的接口测试本文使用了 Postman 作为测试软件，Postman 是一个接口测试工具，Postman 相当于一个客户端，它可以模拟用户发起各类 HTTP 请求，将请求数据发送至服务端，获取对应的相应结果，从而验证响应中的结果数据是否和预期值相匹配；并保证开发人员能够及时处理接口中的 bug。

5.3.1.1 登录接口测试

对服务器的登录接口进行测试的主要测试内容为：

- （1）输入错误密码进行测试，查看是否可以响应失败信息，测试结果如图 5-8 所示。
- （2）输入正确的密码登录成功后，查看是否可以返回 token 字符串，测试结果如图 5-9 所示。

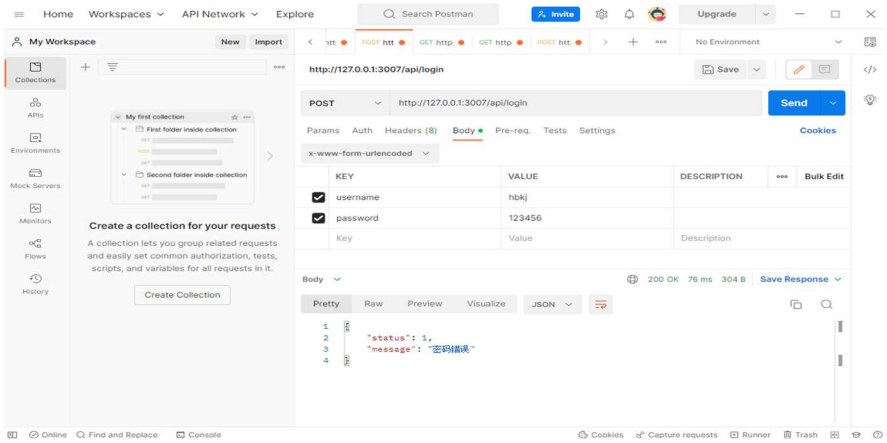


图 5-8 测试 1

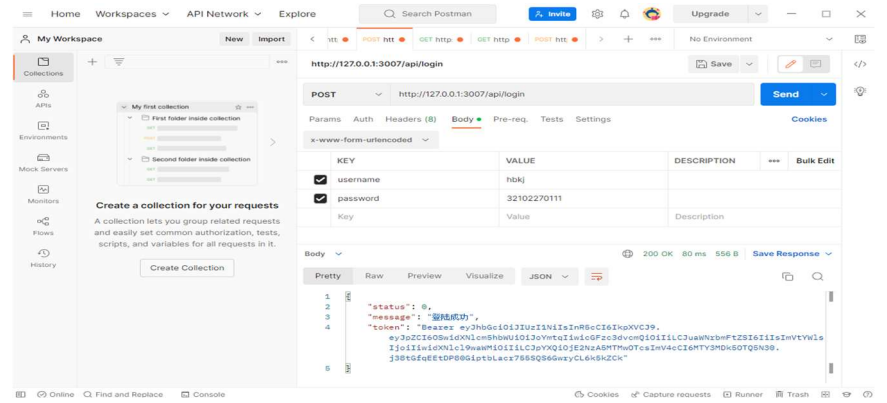


图 5-9 测试 2

5.3.1.2 实时数据接口测试

对实时数据接口进行测试的主要内容为：

- （1）测试在没有获取权限时实时数据接口是否会响应数据，测试结果如图 5-10 所示。
- （2）测试获取权限后实时数据接口是否可以正常响应数据，测试结果如图 5-11 所示。

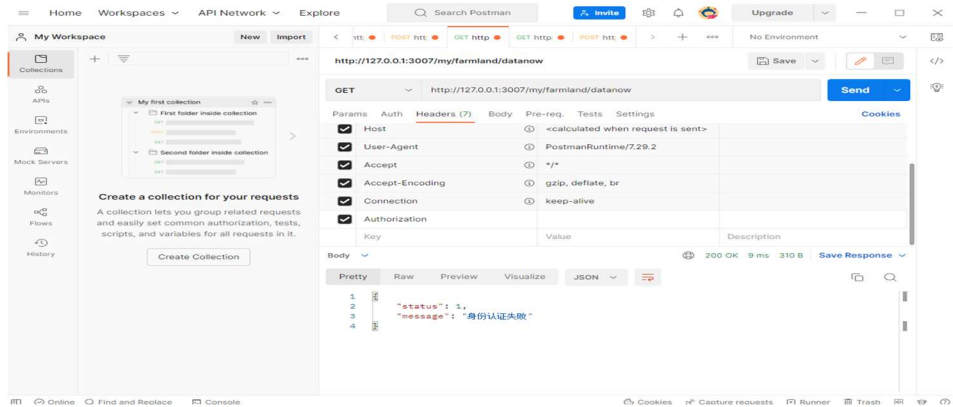


图 5-10 测试 1

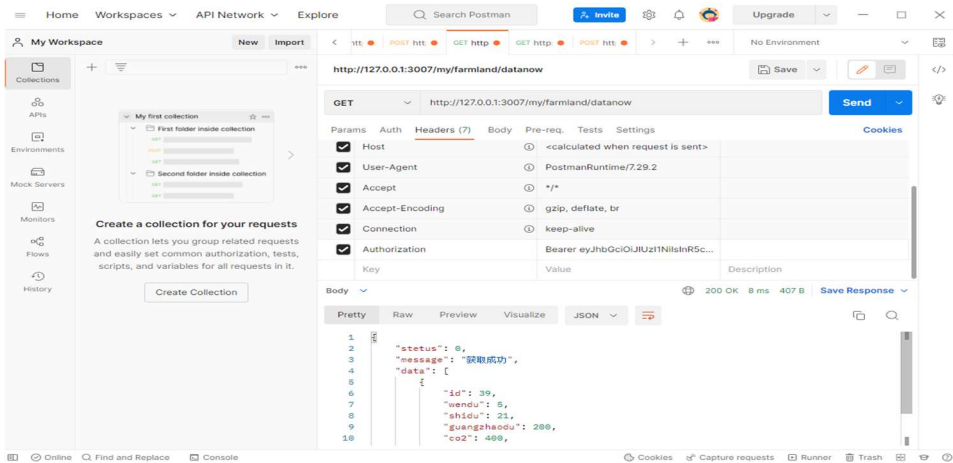


图 5-11 测试 2

5.3.1.3 历史数据接口测试

对历史数据接口进行测试的主要内容为：

- （1）测试在没有获取权限时历史数据接口是否会响应数据，测试结果如图 5-12 所示。
- （2）测试获取权限后历史数据接口是否可以正常响应数据，测试结果如图 5-13 所示。

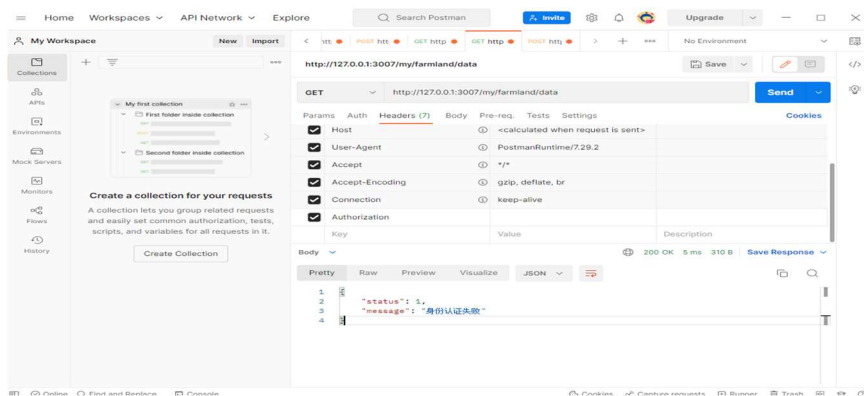


图 5-12 测试 1

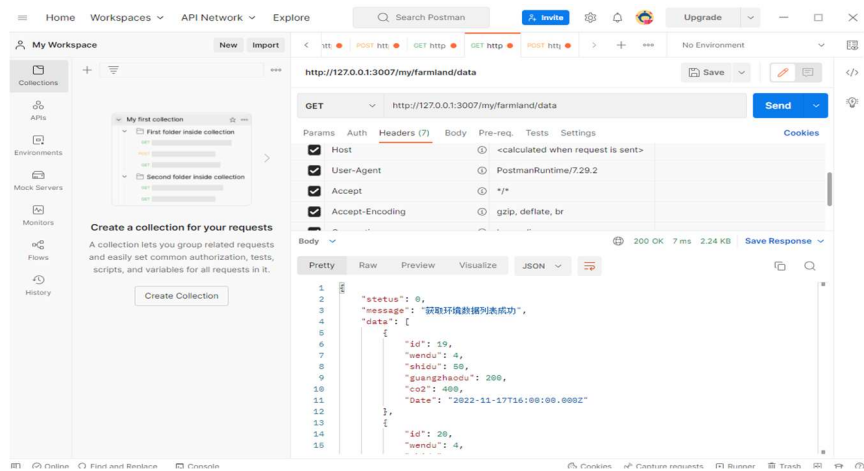


图 5-13 测试 2

根据以上的测试结果可以看出，农田环境监测系统的移动端服务器可以正常运行。

5.3.2 移动端客户端测试

移动端客户端客户端的测试主要是测试客户端的登录功能，实时数据展示功能和历史数据展示功能能否正常运行。

5.3.2.1 登录模块测试

- 对客户端登录模块的测试主要为：
- （1）输入错误的密码是否能接收服务器响应的信息进行错误提示。测试结果如图 5-14 所示。
 - （2）输入正确的密码登录后，是否能接收服务器响应的 token 字符串，测试结果如图 5-15 所示。

(3) 是否可以将 token 字符串进行储存，测试结果如图 5-16 所示。

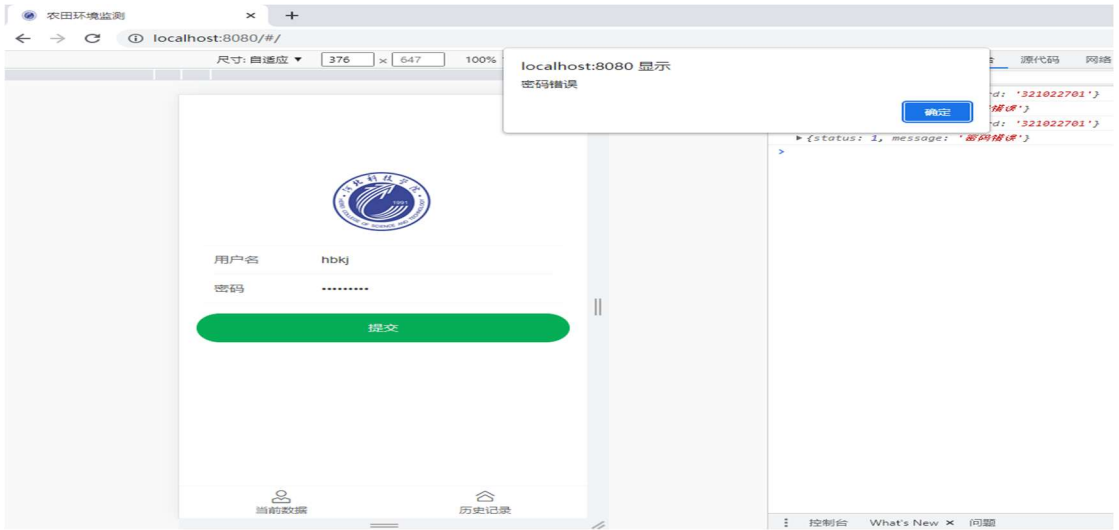


图 5-14 测试 1

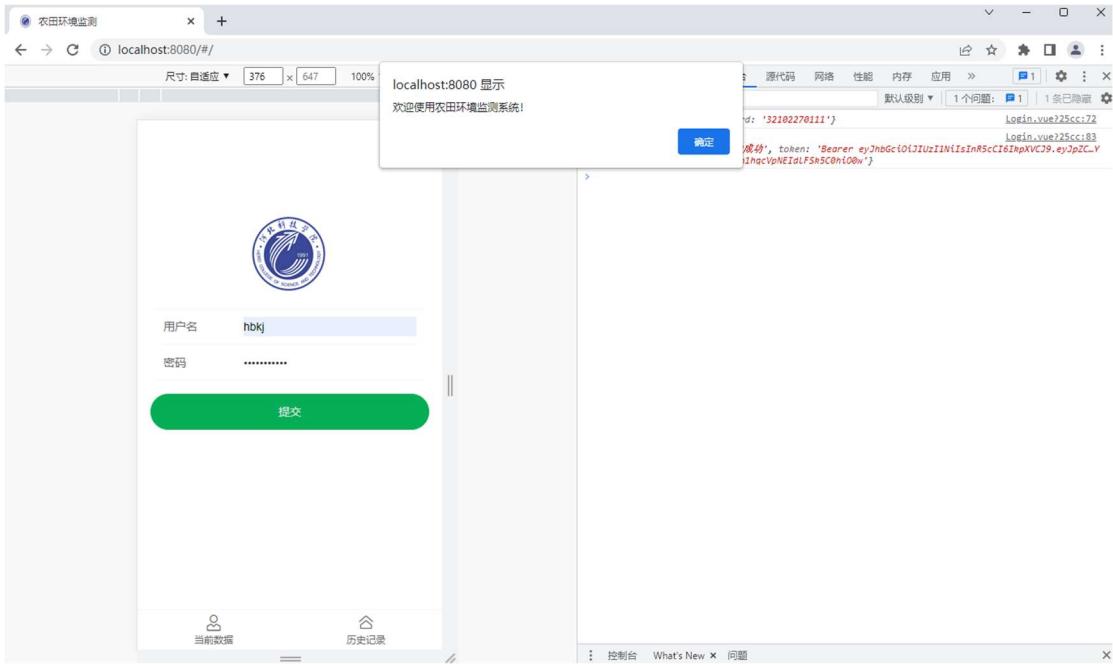


图 5-14 测试 2

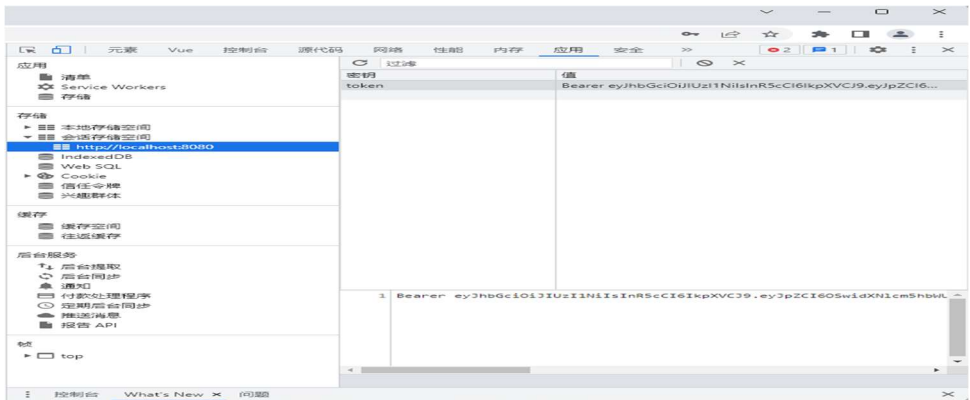


图 5-15 测试 3

5.3.2.2 实时数据展示模块

对客户端实时数据展示模块的测试只要是查看实时数据能否正常接收服务器响应的实时数据，并对是实时数据进行展示。测试结果如图 5-16 所示。

5.3.2.3 历史数据展示模块

对客户端历史数据展示模块的测试只要是查看历史数据能否正常接收服务器响应的历史数据，并对是实时数据进行展示。测试结果如图 5-17 所示。

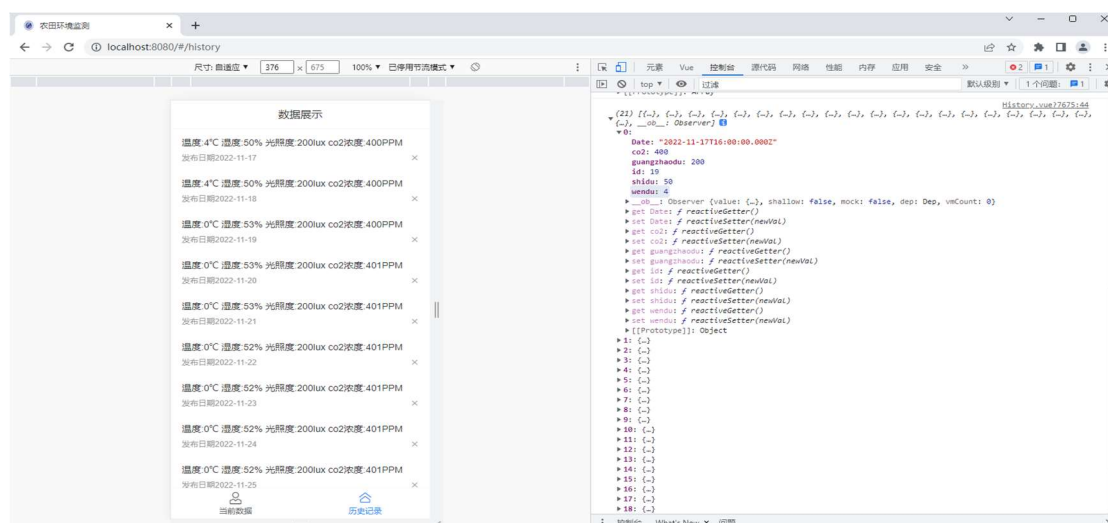


图 5-16 实时数据展示模块测试

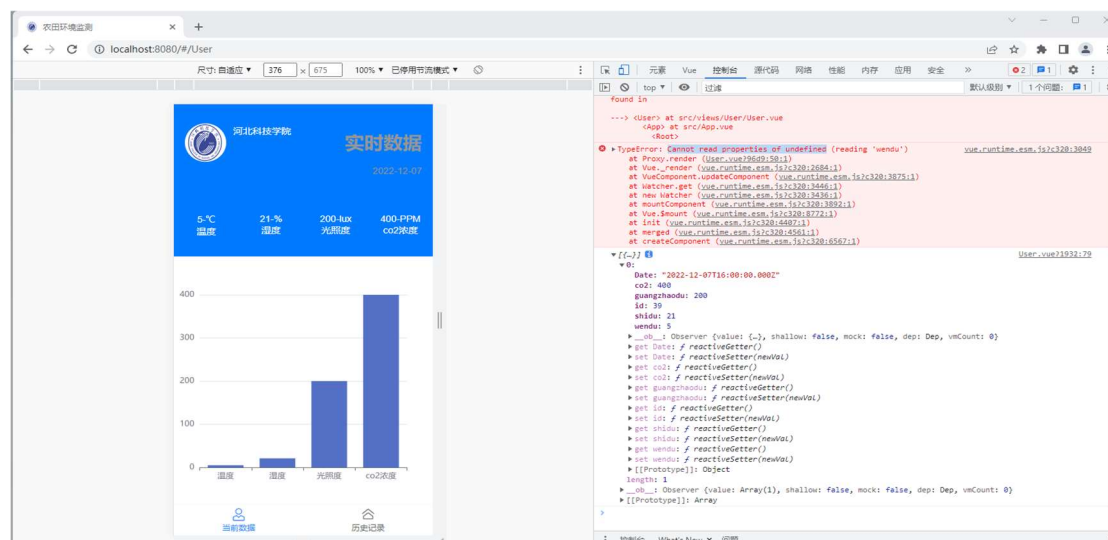


图 5-17 历史数据展示模块测试

在进行实时数据模块与历史数据模块的测试时发现了一个问题，即数据库数据发生变化时，客户端发起 get 请求，请求到的数据没有变化。在查询资料后得

知 `vue` 在多次请求同一个接口时, 不会重复的向服务器发起请求而是直接使用缓存中的数据, 所以会导致数据没有发生变化。解决方法是设置 `axios` 拦截器, `axios` 拦截器分为请求拦截器和响应拦截器。顾名思义, 请求拦截器会在每次发起请求的时候被触发; 响应拦截器会在每次得到响应之后被触发。这里我们设置请求拦截器, 具体功能是可以拦截客户端发送的请求, 并且在完成某一事件后对该请求放行。可以利用请求拦截器的功能, 拦截每一次的 `get` 请求, 并在给每一次的 `get` 请求加上一个时间戳后放行, 这样就不会判定多次调用时调用的同一个接口, 这样就解决了数据库数据变化而响应给客户端的数据不变的问题, 拦截器具体代码如下:

```
axios.interceptors.request.use(function(config) {  
  //在发送请求前给 get 请求加时间戳  
    if (config.method === 'get') {  
      config.params = {  
        t: Date.parse(new Date()) / 1000,  
        ...config.params  
      }  
    }  
    return config  
  },  
  error => {  
    return Promise.reject(error)  
  })
```

在解决上述问题之后农田环境监测系统移动端客户端的测试已经完成, 根据每项测试的结果可以看出农田环境监测系统移动端客户端运行正常。

5.4 本章小结

本章主要介绍了农田环境监测系统的硬件与软件的测试过程与测试结果, 分别对硬件的信息传输、PC 端的各项功能、移动端服务器的各个接口和移动端客户端的各个模块进行了测试。根据每个环节的测试结果来看, 本系统的下位机与上位机可以正常的实现设计的要求。

第 6 章 总结与展望

6.1 总 结

随着通讯设备的复杂度、功耗及成本的增加，相对于其他通讯技术，ZigBee 的低成本、低功耗等诸多优势，在激烈的竞争中脱颖而出。ZigBee 无线通讯技术具有低速率、低成本、低功耗等特点，由于其自身优势，ZigBee 已经被应用到国民经济很多领域，用于高速照明、汽车维修、身份认证、智能家居、火灾报警等，在农业生产方面也有很多应用，可以用其检测农田环境。推动粗放型农业向智能型的精细农业转变，为实时监测田间信息为农业生产提供理论依据。本文主要研究田间数据采集与数据展示的问题，针对这一问题，分析研究了以 ZigBee 技术为基础的无线传感器网络技术，并分析了田间数据的采集、传输功能，并且对系统相应的软、硬件方面进行了设计，其中硬件部分主要对传感器监测区域内的温度、湿度、光照度、CO₂ 浓度进行采集，并将数据传输至协调器由协调器上传至上位机。软件部分主要实现的功能就是获取上位机数据后进行储存与展示。在系统的测试过程中，找出了系统的 bug，并将 bug 进行了修复，修复后经过测试本文所设计的农田环境监测系统运行正常，达到了设计要求。

6.2 展 望

本文主要研究适合田间信息监测的 ZigBee 无线传感器网络，对其进行软、硬件设计。ZigBee 技术作为一种近距离无线通信技术，前景非常广阔，由于时间有限，本文仅仅对田间环境数据进行采集，要想将 ZigBee 技术广泛应用到田间信息监测中，还需要从以下几个方面进行研究：

（1）对田间信息进行监测时，不仅仅只采集本文所述的几个环境因素，还需要对土壤的湿度以及土壤的温度等影响农业生产的参数进行采集，要求接入更多的传感器，进行多路采集，数据采集更加困难。

（2）由于农田监测区域广泛，在农业领域组建大规模的无线传感器网络成为必然趋势，由于无线传感器网络要长时间在恶劣的农田环境中工作，怎样保证系统稳定、可靠、高效成为解决的问题。

参考文献

- [1]董香丽. 基于 ZigBee 无线传感器网络在温室大棚中的应用研究[J]. 科技资讯, 2019, 17(19): 34+36.
- [2]张习博. 基于 ZigBee 路由算法的研究及其在数据采集系统中的研究[J]. 自动化与仪器仪表, 2018(11): 7-10.
- [3]田荣明. 农田环境监测系统在无线传感器网络中的技术分析[J]. 东北农业科学, 2021(5): 117-121.
- [4]唐红霞, 李怀亮. 基于 STM32 的智慧农业系统设计. 电子制作[J], 2019(9): 20.
- [5]胡伟伟, 赵文龙, 程若发, et al. ZigBee 自组织网络在大棚环境监测中的应用[J]. 现代电子技术, 2018, 41(16): 35-38.
- [6]Yongfei Ye; Xinghua Sun; Minghe Liu; Zhisheng Zhao; Xiao Zhang; Hongxi Wu The remote farmland environment monitoring system based on ZigBee sensor network [J] International Journal of Computational Science and Engineering, 2018.
- [7]刘飞飞, 徐隆姬, 马礼然. 基于 ZigBee 的分布式农业环境监测系统设计[J]. 传感器与微系统, 2021(03).
- [8]刘建波, 李红艳, 孙世勋, 杨兴龙. 国外智慧农业的发展经验及其对中国的启示[J]. 世界农业, 2018, (11): 13-16.
- [9]戴珍蕤. 促进我国智慧农业发展的对策研究[D], 浙江海洋大学, 2018.
- [10]刘星, 李晔. ZigBee 技术开发[M]. 清华大学出版社, 2022.
- [11]程琪骥, 王桂兰, 周明亮. 基于 Zigbee 技术的智能家居安防系统设计[J]. 集成电路应用, 2022, 39(10): 258-259.
- [12]李志伟, 东伟, 黄双成. 基于 DHT11 的农业大棚温湿度监控系统设计[J]. 工业仪表与自动化装置, 2021, (01): 39-43.
- [15]杨世权, 张谦述, 周聪. 多路 BH1750 光强检测系统的设计[J]. 太原学院学报(自然科学版), 2018, 36(04): 59-63.
- [16]聂灵风. 基于 ZigBee 技术的温室智能监控系统[D]. 导师: 刘海峰. 陕西科技大学, 2018.
- [17]Wu Daiwen. The Application and Management System of Scientific Research Projects Based on PHP and MySQL[J]. Journal of Interconnection Networks, 2022, 22(Supp02).
- [18]张鹏飞, 王乾, 胡晓冬, 杨明浩, 崔明旺. 基于 Node.js 和 JS 的前后端分离实现[J]. 软件, 2019, 40(04): 11-17.

[19]刘亚茹, 张军. Vue.js 框架在网站前端开发中的研究[J]. 电脑编程技巧与维护, 2022, (01):18-19+39.