**附件5 佛山大学《无线传感器网络与RFID技术课程设计》课程实习（实训、实践）报告**

**佛 山 大 学**

**计算机与人工智能学院**



**《无线传感器网络与RFID技术课程设计》报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 专 业 | 物联网工程 | 学 号 |  |
| 姓 名 |  | 班 级 |  |
| 日 期 |  | 成 绩 |  |

一、目的

1、本设计是在物联网工程专业课程全部修完后进行的一个重要的强化性实践教学环节，在教师指导下由学生独立完成。在学生掌握物联网工程基本原理的基础上，以项目为主体、职业或行业为引导，通过组织学生参加实际项目的设计、履行和管理，在项目实施过程中完成教学任务的过程。在这过程中，由学生根据物联网工程项目不同阶段的任务要求，利用物联网原理与实验技能分析并解决实际问题，在各阶段完成后形成最终的项目解决方案。学生在项目流程的熟悉与项目需求的实现过程中加深自身对物联网整体结构的理解，强化自己的物联网系统设计、实施、管理和协议分析能力。从而达到对学生综合能力的全面培养，为以后从事与物联网相关的工作打下理论基础。

2、掌握无线传感网络节点硬件电路分析：学生将学习如何分析无线传感网络节点的硬件结构与电路设计，包括传感器、通信模块等组件的选择与布局。

锻炼系统设计与开发能力：通过自主选择题目并进行系统设计，学生能够独立进行需求分析、方案设计、硬件搭建、软件编程、系统调试等全过程的工作，提升其综合运用知识解决实际问题的能力。

培养创新意识与实践能力：实验鼓励学生设计具有一定创新性或先进性的功能，培养学生在技术实现和功能扩展方面的创新思维，并将理论知识转化为实际应用，提升其项目开发与工程实施的能力。

二、原理与方案

**实验原理**

本实验基于Zigbee技术设计一个智能家居系统，通过无线传感器网络（WSN）进行环境数据采集，并实现对智能家居设备的控制。以下是该实验的主要原理分析：

### 1. ****Zigbee技术原理****

Zigbee是一种基于IEEE 802.15.4标准的低功耗、低数据传输速率、短距离无线通信协议，适用于传感器节点和智能家居控制系统中。Zigbee网络采用星形、树形和网状三种拓扑结构，其中网状拓扑结构具备自组织、自修复的特点，适合用于大规模分布式传感器网络。

Zigbee通信模块通过射频（RF）信号与其他设备进行无线通信。Zigbee通信系统由三种类型的设备构成：

* **协调器（Coordinator）**：网络的核心节点，负责网络的建立与管理。
* **路由器（Router）**：负责数据转发，并可以扩展网络范围。
* **终端设备（End Device）**：用于采集数据或执行任务的节点，例如温湿度传感器、光照传感器、灯光控制模块等。

### 2. ****无线传感器网络（WSN）原理****

无线传感器网络由多个传感器节点和数据收集器节点组成，传感器节点负责实时采集环境信息，如温度、湿度、光照强度等。数据收集器节点将传感器采集的数据通过无线方式传输到中心系统进行处理和管理。在本实验中，Zigbee技术提供了高效的无线通信能力，确保数据能够实时地从传感器节点传输到控制中心或移动终端。

无线传感器网络中的数据传输通常采用点对点、点对多点的方式，利用数据包交换技术实现不同节点之间的数据通信。传感器节点与Zigbee网络协调器之间的通信通过信道进行，数据以无线电波的形式在不同节点之间传输。

### 3. ****智能家居控制原理****

在智能家居系统中，Zigbee技术通过无线传感器节点采集城市环境的各种信息，如温度、湿度、PM2.5等，这些信息将传送到中央控制系统进行处理和分析。根据设定的条件，控制系统可以对城市设备进行自动控制，如开关灯等。

三、步骤

1.进行Zigbee组网，构建Zigbee网络。将传感器与节点相连，构成Zigbee节点。

2.Zigbee节点连接协调器，将数据传递到协调器。

3.协调器通过网线连接主机，将传感器节点检测到的数据传递到电脑上

4.电脑将获取到的传感器数据在web端显示出来

#### ****步骤1：进行Zigbee组网，构建Zigbee网络****

1. **选择Zigbee模块**：首先选择适合本实验的Zigbee通信模块——ZXBeeLite-CC2530BeeGX，该模块用于无线传输传感器数据。每个Zigbee模块将作为一个网络节点，与其他节点进行通信。

**2、连接传感器到Zigbee节点**：将需要的传感器（如温湿度传感器、光照传感器等）与Zigbee模块连接。传感器将采集城市环境中的相关数据，并通过Zigbee模块将数据转换为无线信号，供其他节点或协调器接收。

**3、配置Zigbee节点**：为每个Zigbee节点（传感器节点）配置唯一的地址，使其能够在网络中被正确识别，并且能够与协调器节点进行有效的通信。可以通过编程工具（如X-CTU）为Zigbee节点设置网络参数，确保其能够加入到Zigbee网络中。

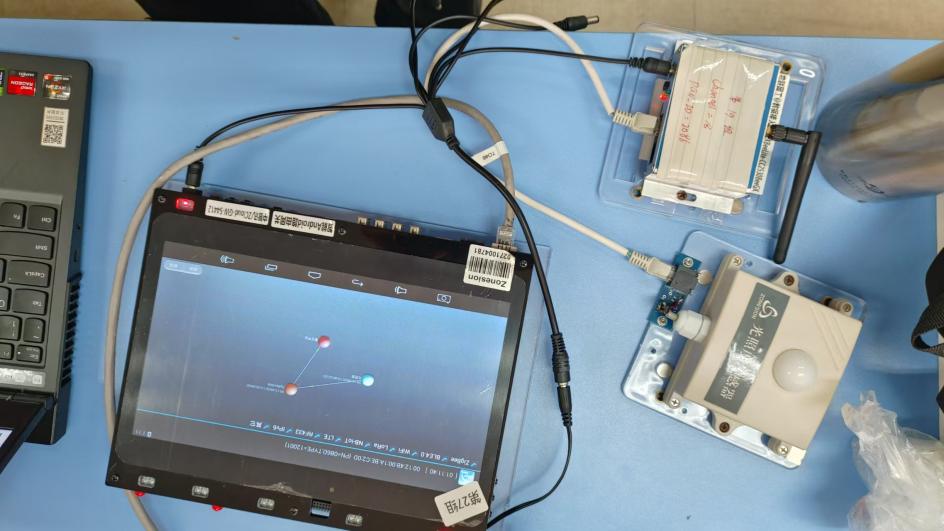
**4、组网拓扑选择**：选择合适的拓扑结构（如星形拓扑、树形拓扑或网状拓扑）来构建Zigbee网络。在本实验中，采用网状拓扑（Mesh Topology），该拓扑能够增加网络的可靠性和覆盖范围，因为节点之间可以多跳转发数据。



#### ****步骤2：Zigbee节点连接协调器，传递数据到协调器****

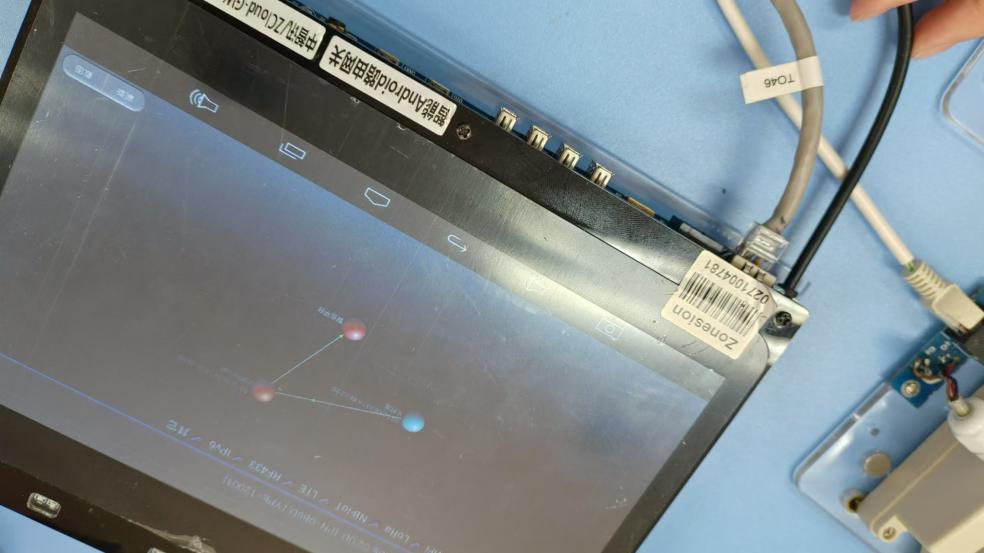
**1、协调器配置**：Zigbee协调器作为网络的中心设备，负责网络的建立与管理，接收来自各个传感器节点的数据。配置协调器节点的通信参数，使其能够与传感器节点进行通信。

**2、节点与协调器通信**：将传感器节点正确配置并加入到Zigbee网络后，它们将通过无线通信将采集到的环境数据如光照度发送到协调器节点。协调器通过Zigbee协议接收数据，并根据需要进行转发处理。



#### ****步骤3：协调器通过网线连接主机，将传感器节点检测到的数据传递到电脑****

1. **连接协调器与主机**：协调器通过网线与计算机进行连接。通过该连接，协调器将数据传递至计算机，计算机可以接收并处理这些数据。



#### ****步骤4：电脑将获取到的传感器数据在Web端显示出来****

**1、编程与数据处理**：处理接收到的传感器数据。程序将从协调器中读取传感器数据，处理并传输至Web端进行显示

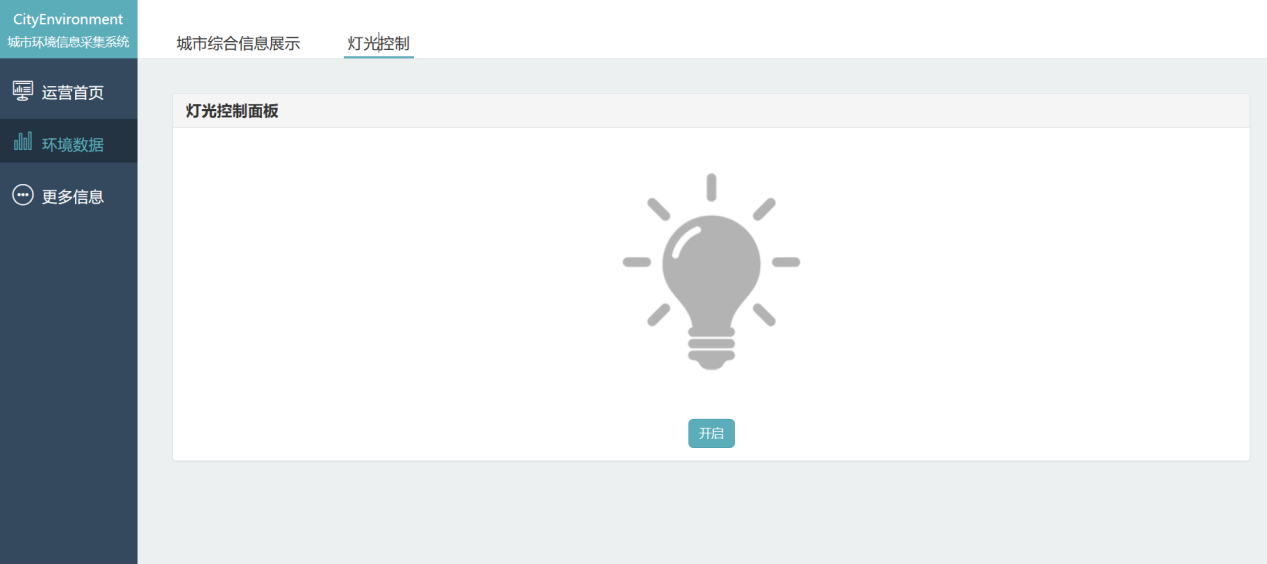
**2、数据传输到Web端**：通过编写Web应用程序，将计算机获取到的传感器数据转发至Web端进行展示。此时，传感器数据会作为实时数据流进行显示。

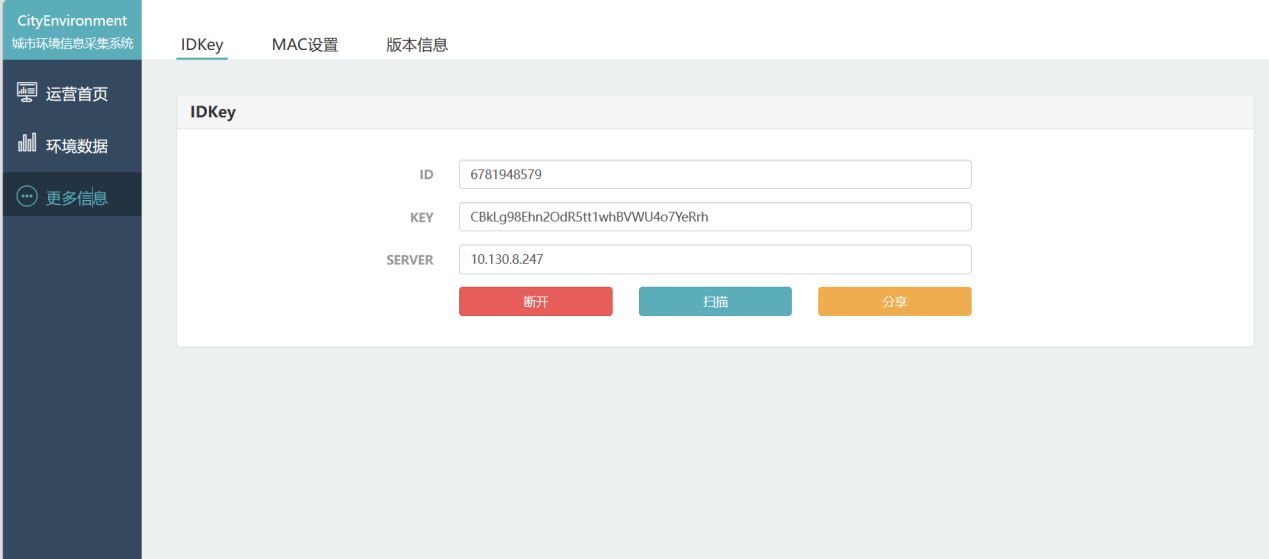
**3、设计Web端显示页面**：在Web端设计用户界面，用于显示传感器数据的实时变化，例如显示当前温度、湿度、光照等环境数据。前端技术可以使用HTML、CSS和JavaScript来实现。

**4、数据可视化**：为了增强用户体验，使用图表库将传感器数据以图表形式展示。图表可以展示实时数据变化趋势，例如，会通过温度计的图表显示数据。

**5、web端控制设备**：用户也可以通过Web端发送命令（如控制灯光）到协调器，从而实现设备的远程控制。







四、数据记录与处理

**1、数据的连接**

function getConnect() {

// 设置连接信息

rtc = new WSNRTConnect(localData['ID'], localData['KEY']);

rtc.setServerAddr(localData['server'] + ":28080");

rtc.connect();

// 连接成功的回调函数

rtc.onConnect = function () {

connectFlag = 1;

message\_show("数据服务连接成功！");

// 处理连接成功后的界面更新

};

// 掉线的回调函数

rtc.onConnectLost = function () {

connectFlag = 0;

message\_show("数据服务连接失败，请输入正确的ID、KEY和server");

};

// 接收消息的回调函数

rtc.onmessageArrive = function (mac, dat) {

// 解析接收到的数据并更新页面

};

}

1. 该函数负责通过 WSNRTConnect 对象建立与远程数据服务的连接，使用 ID、KEY 和服务器地址。
2. rtc.onConnect：连接成功后的处理函数，更新界面并显示成功信息。
3. rtc.onConnectLost：连接丢失后的处理函数，更新界面并显示错误信息。

4、rtc.onmessageArrive：接收到来自数据服务的消息时的回调函数，用于解析和处理接收到的环境数据（如 PM2.5、温湿度、光照强度等）。

1. **消息的处理**

rtc.onmessageArrive = function (mac, dat) {

if (dat[0] == '{' && dat[dat.length - 1] == '}') {

// 解析消息格式

var its = dat.split(',');

for (var x in its) {

var t = its[x].split('=');

if (t.length != 2) continue;

// 根据不同的MAC地址更新不同城市的传感器数据

if (mac == localData.BJ\_PMMAC) {

// 更新北京的传感器数据

dial3('BJ\_CO2', 'ppm', Data);

}

// 处理其他城市的数据

}

}

};

rtc.onmessageArrive：这个回调函数处理接收到的数据。数据格式通常是通过分隔符（如逗号）分隔的键值对。根据不同的MAC地址，系统更新不同城市的传感器数据，并通过图表或仪表盘显示。

1. **数据发送**

$(".mac-input").click(function(){

var currentCity = $(this).parents(".panel").find(".panel-heading").text();

// 更新localData并发送消息

eval('rtc.sendMessage(localData.'+city[currentCity]+'\_PMMAC, "{A0=?,A1=?,A4=?}")');

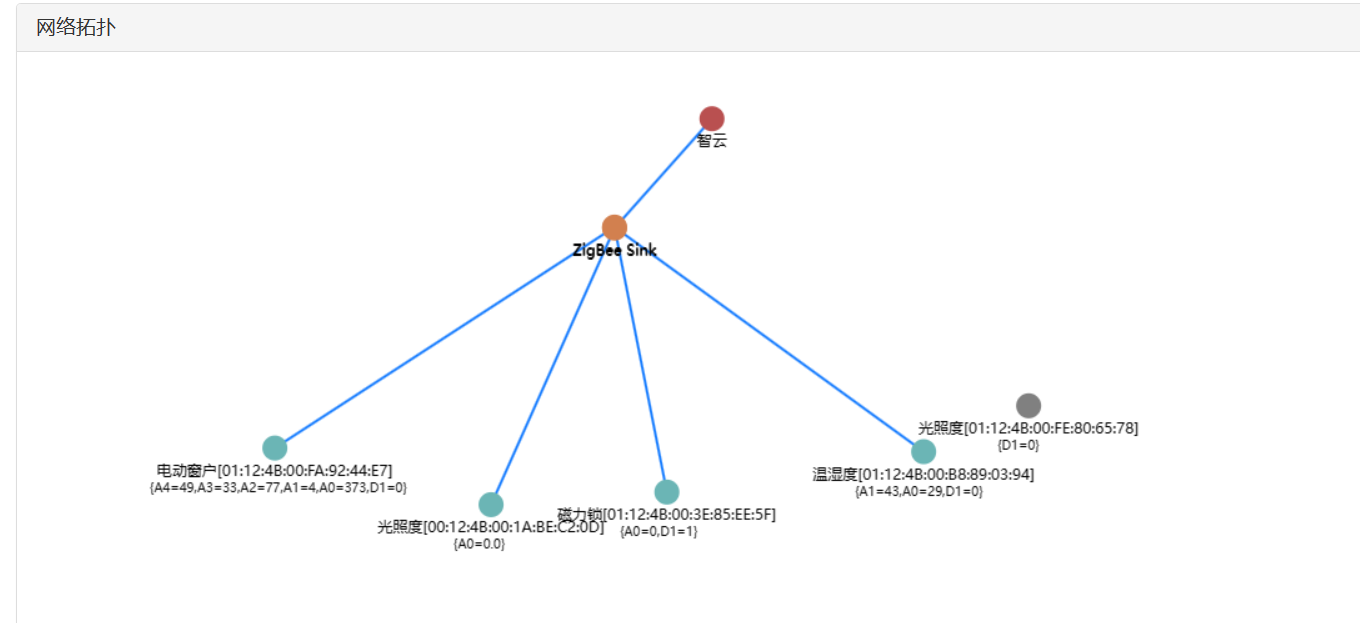
message\_show(currentCity+"MAC设置成功");

});

在点击“MAC输入”按钮后，系统将根据当前城市（北京、上海、深圳、武汉）设置对应的MAC地址，并通过 rtc.sendMessage 向服务器发送命令或请求数据。

五、结果分析

**1、Zigbee网络的拓扑图**



1. web端页面



六、总结与心得

通过参与《智能家居控制》项目，我深刻体会到了智能家居技术的魅力和潜力。项目不仅让我对ZigBee技术有了全面的认识，也锻炼了我的实践能力和创新思维。本项目旨在设计一个基于ZigBee技术的智能家居检测系统，通过无线传感器网络实现家庭环境的实时监控和管理。ZigBee以其低功耗、低成本和高可靠性的特点，成为智能家居领域的理想选择。项目过程中，通过阅读和实际操作，我对ZigBee技术的原理和应用有了更深刻的理解。系统设计阶段，我参与了系统架构的规划和模块的划分。我们采用了模块化设计，确保了系统的灵活性和可扩展性。我们模拟系统时使用了现有的实验室硬件设备和智云硬件仿真平台的模拟设备，选择了合适的传感器和Zigbee节点，模拟设备有空气质量传感器、火焰传感器和温湿度传感器，实体硬件有可燃气体传感器、人体红外探测器和光照度传感器。功能实现上，我们重点考虑了用户交互和数据处理的效率。在实际开发过程中，遇到了不少技术难题，敲代码的过程中也有许多报错和不兼容，数据返回方面也有一些问题出现，通过团队协作和不断调试，我们逐步解决了这些问题。这个过程锻炼了我的问题解决能力和耐心。通过这个项目，我的能力都得到了显著提升。更重要的是，我学会了如何在团队中有效沟通和协作，这对于我未来的职业生涯是非常宝贵的经验。这个项目不仅让我对智能家居技术有了更深入的了解，也让我认识到了技术创新和团队合作的重要性