**中国海洋大学计算机科学与技术学院**

实验报告

**姓名：惠欣宇 年级：2021级 专业：计算机科学与技术**

**科目：物联网系统设计与开发 题目：智能大棚场景仿真设备交互编程**

**实验时间：2024年 5月 23日 实验教师：郭忠文**

**目录**

[一、实验目的及要求 2](#_Toc168775161)

[1. 实验目的 2](#_Toc168775162)

[2. 实验要求 2](#_Toc168775163)

[二、实验内容 2](#_Toc168775164)

[三、实验设备 3](#_Toc168775165)

[四、通信协议 3](#_Toc168775166)

[五、实验步骤 7](#_Toc168775167)

[六、报告要求 11](#_Toc168775168)

[1. 简要叙述实验结果 11](#_Toc168775169)

[2. 把打印采集程序和控制程序源代码和存储数据的文本文件附在实验报告中 11](#_Toc168775170)

[3. 总结编写数据采集软件及软件调试的关键步骤 13](#_Toc168775171)

[4. 写出调试过程遇到的主要问题和解决方法 14](#_Toc168775172)

[七、实验总结 15](#_Toc168775173)

一、实验目的及要求

1. 实验目的

智能农业大棚传感设备既可以采集数据，又可以通过命令控制设备，仿真器与计算机通过网线连接，通过对智能农业大棚传感设备的交互编程，加深对通过混合网络进行传感数据获取和控制设备的认识，为物联网系统综合数据采集软件设计与开发打下基础。编写数据采集软件，利用有线网络和互联网关通信，获取智能大棚传感设备数据，理解智能传感设备的通信协议，以及获取智能农业大棚环境设备的控制方式。加深对通过混合网络进行传感数据获取方法的认识，为物联网系统综合数据采集软件设计与开发打下基础。。

2. 实验要求

预先了解实验环境，理解通信协议，掌握在有线、无线混合网络环境下，计算机与传感设备进行通信的方法。根据实验内容，预先编写数据采集软件，实验过程中认真调试、思考和记录，掌握利用软件自动获取混合网络传感设备数据及控制的方法，撰写实验报告。

二、实验内容

编写数据采集软件，获取仿真器提供智能大棚的传感设备数据，数据采集软件实现如下两种功能：

（1）具有控制功能：实现风机启、停，电灯开、关，水龙头开、关以及二 氧化碳发生器的启动、停止控制。

（2）编写数据采集软件获取智能大棚的土壤温度、湿度、二氧化碳浓度和照度数据，把解析后的数据，写入桌面的文本文件，文件以姓名加学号命名，每次采集的数据在文件中占据一行，每行的开头为采集数据的时间格式为“YYYY-MM-DD,hh:mm:ss”，如：2017-03-26,15:28:32。不同数据项以“;”号分隔，共采集10 次。

三、实验设备

1）智能大棚仿真设备一套；

2）网线一根；

3）计算机一台；

4）USB-网口转换器一块。

四、通信协议

智能大棚硬件结构如图5.4.1所示。计算机与物联网网关建立网络通信，计 算机向物联网网关发送指令，物联网网关将指令通过无线自动转发到传感器设备 或控制模块，传感器设备和控制模块做出响应后将返回的结果传输的互联网网关， 物联网关通过有线网络自动转发到计算机。因此，为获取土壤传感数据或实现有 关设备的控制，计算机只需与物联网关进行通信。

图示

描述已自动生成

**图5.4.1** 智能大棚硬件结构

JZH-026 传感设备及JZH-2控制设备通信协议如下：

1）查询命令 KL-H1100 网关网线 JZH-026 型无线传感设备感知温度、湿度等信息，并存储在寄存器中，每两 个寄存器存放一个传感器的数据，从地址为 0 的寄存器开始依次存放温度、湿 度、照度、土壤温度和CO2浓度的数据。

例如查询传感设备地址为1的前五个传感器的数据命令为：15 01 00 00 00 06 01 03 00 00 00 0A（十六进制数值）。查询命令格式见表5-4-1。

表格, 日历

描述已自动生成

表5-4-1名称中每部分的含义如下：

①事务单元标识符：请求与未来响应建立连接，在此固定为15 01；

②协议标示符：报头之后的协议的协议号，modbusTCP协议号位00 00；

③长度：从第7个字节（设备地址）开始到最后所有数据的字节数；

④设备地址：传感设备的地址，一个物联网网关可以连接多个设备，每个设备的地址不同；

⑤功能码：此命令是查询命令还是控制命令，查询命令的功能码为03；

⑥寄存器起始地址：从哪一个寄存器开始读取数据；

⑦读取寄存器数量：从寄存器起始地址开始读取多少个寄存器的数据；

无线传感设备接收到命令后作出响应，返回数据：15 01 00 00 00 17 01 03 14 01 81 01 02 02 01 00 F9 03 00 00 80 04 81 00 F9 05 00 01 C2，返回数据格式说明见 表5-4-2。

表格

描述已自动生成

表格

描述已自动生成

表5-4-2 名称中每部分的含义如下：

①事务单元标识符：请求与未来响应建立连接，在此固定为15 01；

②协议标示符：报头之后的协议的协议号，modbusTCP协议号位00 00；

③长度：从第7个字节（设备地址）开始到最后所有数据的字节数；

④设备地址：传感设备的地址，一个物联网网关可以连接多个设备，每个 设备的地址不同；

⑤功能码：此命令是查询命令还是控制命令，查询命令的功能码为03；

⑥返回寄存器数据的字节数：返回数据的字节数，一个寄存器占两个字 节，读取10个寄存器，返回20个字节的数据；

⑦通道数据：每一个通道代表一个传感器的数据，一个通道占两个寄存器，四个字节。

2）控制命令

JZH-2 型无线控制有八路开关，可以改变每一路寄存器的数值，从而控制设备开启或关闭， 每两个寄存器存放一路开关的状态，从地址为0的寄存器开始依次对应八路开关。

例如控制地址为1的无线控制模块，打开第一路和第二路通道，关闭第三路 和第四路通道，命令为：15 01 00 00 00 0B 01 10 00 00 00 08 10 A1 40 FF FF A2 40 FF FF A3 40 00 00 A4 40 00 00。控制指令格式见表5-4-3。

表格

描述已自动生成

文本

中度可信度描述已自动生成

表5-4-3 名称中每部分的含义如下：

①事务单元标识符：请求与未来响应建立连接，在此固定为15 01；

②协议标示符：报头之后的协议的协议号，modbusTCP协议号位00 00；

③长度：从第7个字节（设备地址）开始到最后所有数据的字（2字节）数；

④设备地址：传感设备的地址，一个物联网网关可以连接多个设备，每个设备的地址不同；

⑤功能码：此命令是查询命令还是控制命令，控制命令的功能码为10；

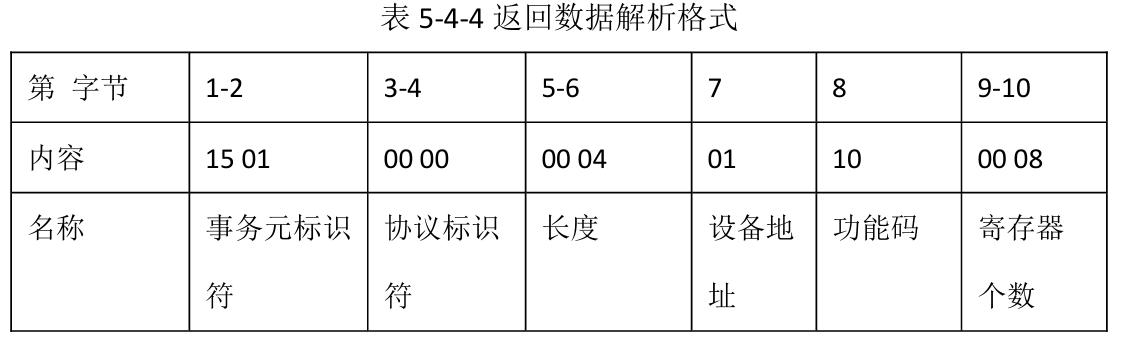
⑥寄存器起始地址：从哪一个寄存器开始控制，将数据写入对应寄存器 中；

⑦控制寄存器数量：从寄存器起始地址开始将数据写入多少个寄存器；

⑧字节数：所有寄存器一共有多少个字节，也就是寄存器数量的两倍；

⑨14至29字节为写入四路开关寄存器的数值。

无线控制模块做出相应控制后，返回:15 01 00 00 00 04 01 10 00 08，解析格 式见表5-4-4：



五、实验步骤

（1）笔记本电脑与智能大棚仿真设备硬件连接方式如图5.4.2所示，对于具有线网络接口的笔记本电脑选择其中图（a）方式，否则选择图（b）方式。

图示

描述已自动生成

（2）打开仿真器桌面上的“琏雾实验系统\实验5.4”文件夹，运行“kl h1100”快捷方式，仿真器屏幕显示智能大棚监测的仿真设备，如图5.4.3所示。

图示

描述已自动生成

**图5.4.3** 智能大棚传感设备仿真界面

（3）在物联网网关上点击右键，屏幕出现互联网关设置界面，如图5.4.4所 示，通过此界面设置IP地址、端口号、信道和网络ID，信道和网络ID都是十六 进制数值，设置完后点击确定，物联网网关作为网络通信的服务端，等待网络调试助手的连接。JZH-026传感器设备和JZH-2网由只需设置信道、网络ID和地址。

图形用户界面

描述已自动生成

**图5.4.4** 互联网关设置界面

（4）打开计算机桌面“琏雾实验系统\工具”文件夹，运行“网络助手”软件，协议类型选择TCP Client，本地主机地址选择实验电脑的IP地址，远程主机 地址输入刚刚第三步设置的物联网网关的 IP 地址和端口号，在网络数据发送区 输入“15 01 00 00 00 06 01 03 00 00 00 0A”（十六进制），点击发送，在网络数据 接收区会显示获取的传感设备数据其中包含了温度、湿度、照度和CO2浓度，确 保硬件通信线路正常。

（5）对根据实验内容编写的程序进行调试。

导入两个标准库。socket库用于创建网络通信接口，通过TCP/IP协议与远程设备进行数据传输。time库用于获取当前时间和暂停程序执行，以便在数据获取和控制命令发送之间插入时间间隔。

1. import socket  # 导入socket库，用于网络通信
2. import time  # 导入time库，用于时间操作

下面是parse\_sensor\_data()函数，这个函数用于解析从传感器设备接收到的二进制响应数据。它将特定字节范围内的数据转换为相应的传感器读数，包括温度、湿度、照度、土壤温度和CO2浓度。解析后的数据以实际单位返回（例如，温度以摄氏度，湿度以百分比表示）。

1. # 解析传感器数据的函数
2. def parse\_sensor\_data(response):
3. temp = int.from\_bytes(response[11:13], byteorder='big') / 10.0  # 从响应中提取并计算温度
4. humidity = int.from\_bytes(response[15:17], byteorder='big') / 10.0  # 从响应中提取并计算湿度
5. light = int.from\_bytes(response[19:21], byteorder='big')  # 从响应中提取照度
6. soil\_temp = int.from\_bytes(response[23:25], byteorder='big') / 10.0  # 从响应中提取并计算土壤温度
7. co2 = int.from\_bytes(response[27:29], byteorder='big')  # 从响应中提取CO2浓度
8. return temp, humidity, light, soil\_temp, co2  # 返回解析后的传感器数据

下面是用于将传感器数据写入指定的文本文件中的函数write\_to\_file()。每次调用函数时，数据都会以追加模式写入文件，从而保留先前的记录。这样可以持续记录一段时间内的传感器读数。

1. # 写入文件的函数
2. def write\_to\_file(data, filename):
3. with open(filename, 'a') as file:  # 以追加模式打开文件
4. file.write(data + '\n')  # 将数据写入文件并换行

下面是根据传感器读数生成相应的控制命令并通过网络发送给设备的函数。根据温度、湿度、照度和CO2浓度的不同值，决定是否开启或关闭风扇、水龙头、灯光和CO2发生器。生成的控制命令以十六进制格式发送到设备，调整其状态。

1. # 发送控制命令的函数
2. def send\_control\_command(temp, humidity, light, solid\_temp, co2):
3. commands = []  # 初始化命令列表
4. if temp > 21.9:  # 如果温度高于21.9°C
5. commands.append('A1 40 FF FF')  # 打开风扇
6. else:
7. commands.append('A1 40 00 00')  # 关闭风扇
8. if humidity > 45.3:  # 如果湿度高于45.3%
9. commands.append('A2 40 00 00')  # 关闭水龙头
10. else:
11. commands.append('A2 40 FF FF')  # 打开水龙头
12. if light > 1900:  # 如果照度高于1900Lux
13. commands.append('A3 40 00 00')  # 关灯
14. else:
15. commands.append('A3 40 FF FF')  # 开灯
16. if co2\_concentration > 425:  # 如果CO2浓度高于425PPm
17. commands.append('A4 40 00 00')  # 关闭CO2发生器
18. else:
19. commands.append('A4 40 FF FF')  # 打开CO2发生器
20. command = '15 01 00 00 00 0B 01 10 00 00 00 08 10'  # 初始化控制命令前缀
21. for i in commands:  # 遍历每个命令
22. command += i.replace(' ', '')  # 去掉命令中的空格并追加到总命令字符串
23. s.sendall(bytes.fromhex(command))  # 发送控制命令

下面是主程序。首先定义与设备通信所需的IP地址和端口号，并指定了数据保存的文件名和查询命令。然后通过创建TCP/IP socket连接，定时（每秒一次）向设备发送查询命令，接收并解析响应数据。解析后的传感器数据包含时间戳并被写入文件，同时根据这些数据生成相应的控制命令以调整设备状态。整个过程循环10次，每次循环之间暂停1秒。

1. # 主函数
2. ip\_address = "192.168.2.3"  # 设备IP地址
3. port = 10005  # 设备端口号
4. filename = "惠欣宇-21030031009.txt"
5. query\_command\_hex = "15 01 00 00 00 06 01 03 00 00 00 0A"  # 查询命令的十六进制字符串
6. query\_command = bytes.fromhex(query\_command\_hex.replace(" ", ""))  # 将十六进制字符串转换为字节
7. with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:  # 创建连接
8. s.connect((ip\_address, port))  # 连接到设备
9. for i in range(10):
10. s.sendall(query\_command)  # 发送查询命令
11. response = s.recv(1024)  # 接收响应数据
12. print(f"返回数据: {response.hex()}")  # 打印返回的数据（十六进制）
13. temp, humidity, light, soil\_temp, co2 = parse\_sensor\_data(response)
14. timestamp = time.strftime("%Y-%m-%d,%H:%M:%S", time.localtime())
15. data\_line = f"{timestamp}; 温度:{temp}°C; 湿度:{humidity}%RH; 照度:{light}Lux; 土壤温度:{soil\_temp}°C; CO2浓度{co2}PPm"
16. print(f"{data\_line}")
17. write\_to\_file(data\_line, filename)  # 将数据行写入文件
18. send\_control\_command(temp, humidity, light, soil\_temp, co2)  # 发送控制命令
19. time.sleep(1)

六、报告要求

1. 简要叙述实验结果

实验实现了智能大棚场景仿真设备的数据采集和控制。通过编写的数据采集软件，我们能够从传感器获取温度、湿度、照度、土壤温度和CO2浓度等数据，并以指定格式将数据记录在文本文件中。同时，根据环境参数的变化，软件能够自动发送控制命令以调整大棚内的设备状态，如开启或关闭风扇、电灯、水龙头和CO2发生器。

2. 把打印采集程序和控制程序源代码和存储数据的文本文件附在实验报告中

**① 源代码**

1. import socket  # 导入socket库，用于网络通信
2. import time  # 导入time库，用于时间操作
3. # 解析传感器数据的函数
4. def parse\_sensor\_data(response):
5. temp = int.from\_bytes(response[11:13], byteorder='big') / 10.0  # 从响应中提取并计算温度
6. humidity = int.from\_bytes(response[15:17], byteorder='big') / 10.0  # 从响应中提取并计算湿度
7. light = int.from\_bytes(response[19:21], byteorder='big')  # 从响应中提取照度
8. soil\_temp = int.from\_bytes(response[23:25], byteorder='big') / 10.0  # 从响应中提取并计算土壤温度
9. co2 = int.from\_bytes(response[27:29], byteorder='big')  # 从响应中提取CO2浓度
10. return temp, humidity, light, soil\_temp, co2  # 返回解析后的传感器数据
11. # 写入文件的函数
12. def write\_to\_file(data, filename):
13. with open(filename, 'a') as file:  # 以追加模式打开文件
14. file.write(data + '\n')  # 将数据写入文件并换行
15. # 发送控制命令的函数
16. def send\_control\_command(temp, humidity, light, solid\_temp, co2):
17. commands = []  # 初始化命令列表
18. if temp > 21.9:  # 如果温度高于21.9°C
19. commands.append('A1 40 FF FF')  # 打开风扇
20. else:
21. commands.append('A1 40 00 00')  # 关闭风扇
22. if humidity > 45.3:  # 如果湿度高于45.3%
23. commands.append('A2 40 00 00')  # 关闭水龙头
24. else:
25. commands.append('A2 40 FF FF')  # 打开水龙头
26. if light > 1900:  # 如果照度高于1900Lux
27. commands.append('A3 40 00 00')  # 关灯
28. else:
29. commands.append('A3 40 FF FF')  # 开灯
30. if co2\_concentration > 425:  # 如果CO2浓度高于425PPm
31. commands.append('A4 40 00 00')  # 关闭CO2发生器
32. else:
33. commands.append('A4 40 FF FF')  # 打开CO2发生器
34. command = '15 01 00 00 00 0B 01 10 00 00 00 08 10'  # 初始化控制命令前缀
35. for i in commands:  # 遍历每个命令
36. command += i.replace(' ', '')  # 去掉命令中的空格并追加到总命令字符串
37. s.sendall(bytes.fromhex(command))  # 发送控制命令
38. # 主函数
39. ip\_address = "192.168.2.3"  # 设备IP地址
40. port = 10005  # 设备端口号
41. filename = "惠欣宇-21030031009.txt"
42. query\_command\_hex = "15 01 00 00 00 06 01 03 00 00 00 0A"  # 查询命令的十六进制字符串
43. query\_command = bytes.fromhex(query\_command\_hex.replace(" ", ""))  # 将十六进制字符串转换为字节
44. with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:  # 创建连接
45. s.connect((ip\_address, port))  # 连接到设备
46. for i in range(10):
47. s.sendall(query\_command)  # 发送查询命令
48. response = s.recv(1024)  # 接收响应数据
49. print(f"返回数据: {response.hex()}")  # 打印返回的数据（十六进制）
50. temp, humidity, light, soil\_temp, co2 = parse\_sensor\_data(response)
51. timestamp = time.strftime("%Y-%m-%d,%H:%M:%S", time.localtime())  # 获取当前时间并格式化
52. data\_line = f"{timestamp}; 温度:{temp}°C; 湿度:{humidity}%RH; 照度:{light}Lux; 土壤温度:{soil\_temp}°C; CO2浓度{co2}PPm"
53. print(f"{data\_line}")
54. write\_to\_file(data\_line, filename)  # 将数据行写入文件
55. send\_control\_command(temp, humidity, light, soil\_temp, co2)  # 发送控制命令
56. time.sleep(1)

**② 存储数据的文本文件**

2024-06-06,10:13:43; 温度:22.0°C; 湿度:45.3%RH; 照度:1895Lux; 土壤温度:25.2°C; CO2浓度415PPm  
2024-06-06,10:13:44; 温度:22.0°C; 湿度:45.3%RH; 照度:1896Lux; 土壤温度:25.2°C; CO2浓度418PPm  
2024-06-06,10:13:45; 温度:21.9°C; 湿度:45.3%RH; 照度:1897Lux; 土壤温度:25.2°C; CO2浓度421PPm  
2024-06-06,10:13:46; 温度:21.9°C; 湿度:45.3%RH; 照度:1898Lux; 土壤温度:25.2°C; CO2浓度424PPm  
2024-06-06,10:13:47; 温度:21.9°C; 湿度:45.3%RH; 照度:1899Lux; 土壤温度:25.2°C; CO2浓度427PPm  
2024-06-06,10:13:48; 温度:21.9°C; 湿度:45.3%RH; 照度:1900Lux; 土壤温度:25.2°C; CO2浓度430PPm  
2024-06-06,10:13:49; 温度:21.9°C; 湿度:45.3%RH; 照度:1901Lux; 土壤温度:25.2°C; CO2浓度433PPm  
2024-06-06,10:13:50; 温度:21.9°C; 湿度:45.3%RH; 照度:1902Lux; 土壤温度:25.2°C; CO2浓度436PPm  
2024-06-06,10:13:51; 温度:21.9°C; 湿度:45.3%RH; 照度:1903Lux; 土壤温度:25.2°C; CO2浓度439PPm  
2024-06-06,10:13:52; 温度:21.9°C; 湿度:45.3%RH; 照度:1904Lux; 土壤温度:25.2°C; CO2浓度442PPm  
2024-06-06,10:18:04; 温度:22.0°C; 湿度:45.3%RH; 照度:1895Lux; 土壤温度:25.2°C; CO2浓度415PPm  
2024-06-06,10:18:05; 温度:22.0°C; 湿度:45.3%RH; 照度:1896Lux; 土壤温度:25.2°C; CO2浓度418PPm  
2024-06-06,10:18:06; 温度:21.9°C; 湿度:45.3%RH; 照度:1897Lux; 土壤温度:25.2°C; CO2浓度421PPm  
2024-06-06,10:18:07; 温度:21.9°C; 湿度:45.3%RH; 照度:1898Lux; 土壤温度:25.2°C; CO2浓度424PPm  
2024-06-06,10:18:08; 温度:21.9°C; 湿度:45.3%RH; 照度:1899Lux; 土壤温度:25.2°C; CO2浓度427PPm  
2024-06-06,10:18:09; 温度:21.9°C; 湿度:45.3%RH; 照度:1900Lux; 土壤温度:25.2°C; CO2浓度430PPm  
2024-06-06,10:18:10; 温度:21.9°C; 湿度:45.3%RH; 照度:1901Lux; 土壤温度:25.2°C; CO2浓度433PPm  
2024-06-06,10:18:11; 温度:21.9°C; 湿度:45.3%RH; 照度:1902Lux; 土壤温度:25.2°C; CO2浓度436PPm  
2024-06-06,10:18:12; 温度:21.9°C; 湿度:45.3%RH; 照度:1903Lux; 土壤温度:25.2°C; CO2浓度439PPm  
2024-06-06,10:18:13; 温度:21.9°C; 湿度:45.3%RH; 照度:1904Lux; 土壤温度:25.2°C; CO2浓度442PPm

3. 总结编写数据采集软件及软件调试的关键步骤

* **理解通信协议**

我们需要深入理解了Modbus TCP协议的工作原理和数据格式，这是实现数据采集和设备控制的基础。

* **编写数据采集程序**

使用socket库来实现与物联网网关的TCP/IP通信。程序包括以下几个关键部分：

**①连接建立：**通过指定IP地址和端口号，建立与物联网网关的连接。

**②数据查询：**构造并发送查询命令，从传感器设备获取数据。

**③数据解析：**定义parse\_sensor\_data函数，将接收到的二进制数据转换为可读的传感器读数。

**④实现数据存储：**设计了write\_to\_file函数，将解析后的数据按照指定格式写入文本文件，以实现数据的持久化存储。

**⑤编写控制逻辑：**根据智能大棚的环境参数阈值，编写send\_control\_command函数，实现了设备的自动控制逻辑。

4. 写出调试过程遇到的主要问题和解决方法

* **问题①：无法建立网络连接**

问题：程序尝试连接物联网网关时失败，无法发送或接收数据。

解决方法：检查网络配置，包括IP地址、端口号是否正确，网线连接是否稳定。使用网络诊断工具（如ping）测试网络连通性。确认物联网网关服务端是否正常运行，并检查防火墙设置是否允许通信。

* **问题②：查询命令发送失败**

问题：即使建立了连接，发送的查询命令没有得到预期的响应。

解决方法：核查发送的查询命令格式是否正确，包括事务单元标识符、协议标示符、长度、设备地址、功能码等是否符合Modbus TCP协议要求。使用十六进制编辑器检查命令的字节序列。

* **问题③：数据解析不准确**

问题：从传感器接收到的数据无法正确解析为温度、湿度等环境参数。

解决方法：检查数据解析函数parse\_sensor\_data中的字节序（big-endian或little-endian）和数据类型转换是否正确。调整字节索引，确保从正确的字节范围内提取数据，并进行正确的数学运算。

* **问题④：控制命令执行无效**

问题：发送控制命令后，智能大棚内的设备状态没有发生预期的变化。

解决方法：验证控制命令的格式和逻辑，确保寄存器地址和写入值正确。检查控制命令是否正确地反映了设备的开关状态。使用调试输出来跟踪命令发送过程。

七、实验总结

通过本次实验，我深入了解了物联网系统中传感器数据的采集和设备控制的原理和方法。实验过程中，我学习了如何使用Python语言进行网络通信和数据处理，还学习了如何根据实际环境参数编写自动化控制逻辑。在调试过程中，我遇到了数据解析和控制命令执行的问题，但通过仔细分析和调整，最终实现了预期的功能。这次实验不仅加深了我对物联网系统设计和开发的理解，也提高了我的解决实际问题的能力。这次实验不仅让我对物联网技术有了更加深入的了解，也为我未来的学习和研究打下了坚实的基础。