佛山大学《物联网通信技术课程》实验报告一

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | ZStack星状组网 | | | 成绩 |  |
| 专业班级 | [22物联网工程2班] | | |
| 姓名 |  | 学号 |  | 学期 | 2024-2025（1） |
| 指导教师 | 陆海波 | 实验性质 | 验证性 | 实验日期 | 2024.12.3 |
| 一、实验目的 | | | | | |
| （1）熟悉Z-Stack协议栈的源文件架构  （2）熟悉Z-Stack常见接口函数调用  （3）熟悉基于Z-Stack的无线通信  （4）熟悉ZigBee网络拓扑结构 | | | | | |
| 二、实验内容   1. **初始化ZigBee协调器：** 配置ZigBee网络的PAN ID，并确保协调器能够顺利启动，为网络建立基础。 2. **终端节点加入网络：** 配置三个终端节点，使它们能够自动识别并加入由协调器创建的ZigBee网络。 3. **数据通信监控：** 利用串口调试助手实时监控各节点间的数据交互，确保点对点传输功能的准确性。 4. **程序烧录与稳定性测试：** 将编译好的程序烧录到CC2530芯片中，并进行星状网络节点通信的稳定性测试。 5. **任务事件触发分析：** 在Z-Stack框架内，对任务事件触发机制进行调试和分析，以理解其实现细节。 | | | | | |
|  | | | | | |
| 三、实验结果 | | | | | |
| **问题分析**  在实验中，出现了以下问题：  PAN ID设置错误可能导致节点无法加入网络；  终端节点与协调器的通信失败可能源于信道配置不一致；  在串口调试中可能会遇到数据帧解析问题。  **算法**  **节点初始化**： 使用ZDO\_INIT和ZDApp\_StartJoiningCycle函数，实现网络的初始化和节点的自动接入。  **事件触发**： 调用osal\_start\_timerEx()设置软件定时器，通过osal\_set\_event()触发事件，确保任务轮询机制正常运行。  **源代码**  以下是协调器的初始化代码片段：  void SampleApp\_Init( uint8 task\_id ) {  SampleApp\_TaskID = task\_id;  ZDOInit(); // 初始化协调器  SampleApp\_NwkState = DEV\_INIT;  ZDApp\_StartJoiningCycle( SampleApp\_TaskID ); // 启动网络  }  **运行结果**   1. PAN ID设置为0x1234后，协调器成功启动网络； 2. 终端节点依次接入网络，16位短地址分别为0x0001、0x0002和0x0003； 3. 串口调试助手显示，终端节点与协调器之间能够成功收发数据。       **实验结果分析**  在实验中，我们成功构建了ZigBee星型网络，并证实了网络协调器与各个终端节点之间的通信可靠性。在事件触发机制的测试中，任务事件的触发时机与我们的预期完全吻合。 | | | | | |
| 四、实验总结 | | | | | |
| 在本次实验中，我构建了一个基于Z-Stack协议的ZigBee星型网络，深入了解了该协议栈的结构和如何使用其常用接口。实验让我认识到，PAN ID和信道设置对于网络的稳定性极为关键，PAN ID的冲突可能会导致实验结果截然不同。此外，OSAL中的任务和事件触发机制是掌握协议栈运作的核心，为我提供了深入理解的起点。我还发现，通过串口调试助手和可视化工具可以有效地验证节点间的数据传输效果。通过这次实验，我不仅学习了ZigBee网络的基础知识和实现技术，还学会了如何在实际应用中配置和调试网络结构。 | | | | | |