



# 信息工程系

# 教

# 案

课程名称：无线网络应用  
总学时： 54

# 第一周：WSN 概述（9月8日 - 9月13日）

## 1. 教学课题和内容

课题：无线传感器网络（WSN）基础认知

内容：①WSN 定义与核心特征（分布式、自组织、低功耗）；②WSN 系统组成（传感器节点、汇聚节点、管理节点）；③WSN 典型应用场景（智慧农业环境监测、工业设备状态监控、森林防火）；④实验：WSN 硬件模块认知（观察传感器节点、汇聚节点实物，识别核心元件如 CC2530 芯片、天线模块）。

## 2. 教学课时

总学时 2 学时（理论 1 学时 + 实验 1 学时）

## 3. 学生已有知识分析

已有基础：学生已学习《物联网基础》，了解物联网“感知层 - 网络层 - 应用层”体系结构，熟悉传感器（如温度、湿度传感器）的基本功能；掌握单片机（如 51 单片机）的简单操作，对“无线通信”有初步概念（如 WiFi、蓝牙）。

知识缺口：未接触“低功耗、大规模组网”的 WSN 技术，对 WSN 的系统架构、核心协议（如 Zigbee）及硬件模块（如 CC2530 节点）完全陌生；缺乏“分布式网络”的思维，难以理解 WSN “自组织组网”的特点。

## 4. 教学目标

知识目标：理解 WSN 的定义与核心特征，能说出 WSN 的三大组成部分，列举 2 个以上 WSN 典型应用场景。

能力目标：能识别 WSN 硬件模块（传感器节点、汇聚节点），能描述节点上核心元件（芯片、天线）的作用。

素养目标：激发对无线网络技术的探究兴趣，初步建立“技术服务实际需求”的工程思维。

## 5. 教学重点

WSN 的系统组成（传感器节点、汇聚节点、管理节点的功能分工）；

WSN 的典型应用场景（结合物联网专业方向，侧重智慧农业、工业监测）。

## 6. 教学难点

WSN 与常见无线网络（WiFi、蓝牙）的区别（从功耗、组网规模、传输速率角度对比）；  
理解 WSN “自组织” 特性（无需人工干预，节点自动形成网络）。

## 7. 课程思政

展示我国在 WSN 领域的应用成果（如新疆棉花种植基地的 WSN 土壤墒情监测系统），说明 WSN 对农业现代化、节能减排的推动作用，培养学生的科技自信与产业使命感；

强调 WSN “分布式协作” 特性，类比团队工作中 “分工协作、优势互补” 的重要性，引导学生树立团队意识。

## 8. 教学方法

理论教学：讲授法（结合 PPT 展示 WSN 系统架构图）、案例分析法（播放智慧农业 WSN 应用视频）；

实验教学：实物演示法（展示 WSN 节点实物，拆解讲解元件）、分组观察法（4 人一组，轮流观察节点细节，记录元件名称）。

## 9. 教学程序

### （1）导入（15 分钟）

案例导入：播放 “智慧大棚 WSN 监测系统” 视频（传感器节点实时采集温度、湿度，数据无线传输到中控屏），提问：“视频中没有网线，数据如何实现远距离传输？这种网络和我们手机用的 WiFi 有什么不同？”

目标呈现：明确本节课需掌握 “WSN 是什么、由什么组成、能做什么” 三个核心问题。

### （2）教授（45 分钟）

理论讲解（25 分钟）：

WSN 定义：强调 “由大量传感器节点组成，通过无线通信实现数据采集与传输的分布式网络”；

核心特征：对比 WiFi（高功耗、短距离），突出 WSN “低功耗（节点续航 1-2 年）、自组织（节点故障后自动重新组网）、大规模（可容纳上千节点）”；

系统组成：①传感器节点（采集数据，如温湿度）；②汇聚节点（接收节点数据，转发到管理节点）；③管理节点（数据处理与展示，如电脑端软件）；

应用场景：结合物联网专业，重点讲解智慧农业（监测土壤湿度，自动灌溉）、工业监测（设备振动监测，预防故障）。

实验操作（20 分钟）：

教师演示：展示 WSN 传感器节点（含 CC2530 芯片、DHT11 温湿度传感器、天线），标注各元件位置，说明“CC2530 是核心，负责数据处理与无线传输”；

学生实操：分组观察节点实物，填写“WSN 节点元件记录表”（记录元件名称、外观特征、推测功能），教师巡视指导，纠正错误认知（如将天线误认为“指示灯”）。

### **(3) 总结 (10 分钟)**

思维导图梳理：WSN 定义→核心特征→系统组成→应用场景；

对比表格：WSN 与 WiFi、蓝牙的关键参数（功耗、组网规模、传输速率），强化差异认知。

### **(4) 作业**

基础题：绘制 WSN 系统组成图，标注各部分名称及功能；

拓展题：调研身边的 WSN 应用（如小区智能停车、校园路灯监控），记录该应用中传感器节点的采集内容。

### **(5) 反思**

预判难点：学生对“自组织”特性理解较抽象，下次课可通过“模拟节点故障”实验（移除一个节点，观察其他节点是否重新连接）强化认知；

改进方向：增加 WSN 与 WiFi 的实物对比（展示 WSN 节点与 WiFi 模块的大小、功耗标签），让差异更直观。

## **第二周：Zigbee 概述与 IAR 平台使用 (9 月 15 日 - 9 月 26 日)**

### **1. 教学课题和内容**

课题：Zigbee 协议基础与 IAR 开发环境搭建

内容：① Zigbee 协议定义与核心优势（低功耗、低速率、低成本）；② Zigbee 协议栈结构（物理层、MAC 层、网络层、应用层）；③ IAR Embedded Workbench 安装与配置（适配 CC2530 芯片）；④ 实验：IAR 平台实操（创建新项目、添加头文件、设置编译参数，生成 HEX 文件）。

### **2. 教学课时**

总学时 4 学时（理论 2 学时 + 实验 2 学时）

### **3. 学生已有知识分析**

**已有基础：**学生已掌握 WSN 的系统架构，了解“无线通信需要协议支持”；具备计算机基础操作能力（如软件安装、文件管理）；部分学生接触过 Keil（51 单片机开发软件），对“项目创建、编译”有初步概念。

**知识缺口：**未接触 Zigbee 协议，不理解“协议栈”的分层逻辑；IAR 平台为新工具，与 Keil 操作差异大，需从零学习；对“CC2530 芯片与 Zigbee 的关联”认知空白。

## 4. 教学目标

**知识目标：**理解 Zigbee 协议的核心优势，能说出 Zigbee 协议栈的 4 个主要层级，掌握 IAR 平台的基本操作流程。

**能力目标：**能独立完成 IAR 的安装与配置，能创建 CC2530 相关新项目，成功编译生成 HEX 文件。

**素养目标：**培养“按步骤操作软件”的严谨态度，通过协议学习理解“标准化”对技术兼容性的重要性。

## 5. 教学重点

Zigbee 协议的核心优势（低功耗、低成本、支持多节点组网）；

IAR 平台的操作流程（安装→创建项目→添加文件→编译→生成 HEX）。

## 6. 教学难点

Zigbee 协议栈的分层逻辑（各层级的功能分工，如物理层负责信号发送，应用层负责数据格式定义）；

IAR 平台的配置细节（如选择“CC2530F256”芯片型号、设置堆栈大小，避免编译报错）。

## 7. 课程思政

讲解 Zigbee 协议的“开源特性”，说明标准化协议对“不同厂家设备兼容”的重要性（如小米、华为的 Zigbee 设备可互通），引导学生树立“开放协作、遵守标准”的行业意识；

实验中强调“按步骤操作”，若跳过配置步骤会导致编译失败，类比“生活中遵守规则的重要性”，培养学生的规则意识与耐心。

## 8. 教学方法

理论教学：讲授法（结合协议栈分层图）、对比分析法（Zigbee 与 WiFi 协议的参数对比）；

实验教学：演示教学法（教师投屏操作 IAR 安装与项目创建）、分步指导法（将操作拆分为“安装→创建→编译”3 步，每步后检查学生进度）、小组互助法（会 Keil 的学生帮助零基础学生）。

## 9. 教学程序

### (1) 导入 (20 分钟)

问题导入：“上节课我们知道 WSN 需要无线通信，那传感器节点之间用什么‘语言’交流？为什么不用 WiFi？”引出 Zigbee 协议；

旧知衔接：回顾 WSN “低功耗”需求，指出“Zigbee 协议专为低功耗、低速率场景设计，是 WSN 的核心协议之一”。

### (2) 教授 (100 分钟)

理论讲解 (40 分钟)：

Zigbee 协议定义：强调“基于 IEEE 802.15.4 标准的低功耗无线通信协议，适用于短距离、低速率的物联网场景”；

核心优势：①低功耗（节点休眠时电流  $< 1\mu\text{A}$ ）；②低成本（芯片价格  $< 5$  美元）；③多节点（一个网络可容纳 65535 个节点）；④高可靠（支持重传机制）；

协议栈结构：①物理层 (PHY)：处理无线信号发送 / 接收；②MAC 层：负责节点地址分配、信道接入；③网络层 (NWK)：实现组网与路由；④应用层 (APL)：定义数据格式与应用场景（如智能家居控制）；

与 WSN 的关联：说明“Zigbee 是 WSN 的主流协议，CC2530 芯片是支持 Zigbee 的核心硬件”。

实验操作 (60 分钟)：

IAR 安装与配置 (25 分钟)：①安装包解压（注意路径不含中文）；②选择“CC2530F256”芯片型号；③配置编译器（设置“Optimization”为“Low”，避免代码优化导致错误）；

项目创建 (20 分钟)：①新建项目 (Project→Create New Project)；②添加 CC2530 头文件（如“ioCC2530.h”）；③编写简单代码（如“点亮 LED”的初始化代码）；

编译与生成 HEX (15 分钟)：①点击“Make”按钮编译，查看“Build Log”是否有报错（如头文件缺失，指导学生检查文件路径）；②生成 HEX 文件 (Project→Options→Output→勾选“Generate HEX file”)，找到文件保存位置。

### (3) 总结 (15 分钟)

流程梳理：Zigbee 协议（语言）→CC2530 芯片（硬件）→IAR 平台（开发工具）的关联逻辑；

操作要点：IAR 安装“路径无中文”、芯片型号“选对 CC2530”、编译“先查报错再调试”。

### (4) 作业

基础题：绘制 Zigbee 协议栈分层图，标注各层级功能；

实操题：重新安装 IAR 平台，创建新项目，编写“定义一个变量并赋值”的简单代码，编译生成 HEX 文件，提交 HEX 文件路径截图。

## (5) 反思

学生易在“头文件添加”环节出错（路径设置错误），下次课需提前准备“头文件包”，统一放在 D 盘根目录，减少操作障碍；

部分学生对“协议栈分层”理解困难，可类比“快递运输”（物理层 = 运输车，MAC 层 = 快递单号，网络层 = 路由规划，应用层 = 收件人信息），用生活案例简化逻辑。

## 第三周：CC2530 外部中断学习（9 月 22 日 - 9 月 26 日）

### 1. 教学课题和内容

课题：CC2530 外部中断原理与实操

内容：①CC2530 外部中断引脚（P0 口、P1 口、P2 口）与中断触发方式（上升沿、下降沿、双边沿）；②外部中断配置步骤（引脚初始化、中断使能、中断服务函数编写）；③实验：基于 CC2530 外部中断的按键控制（按下按键，触发中断，点亮 LED 灯；松开按键，触发中断，熄灭 LED 灯）。

### 2. 教学课时

总学时 2 学时（理论 1 学时 + 实验 1 学时）

### 3. 学生已有知识分析

已有基础：学生已熟悉 CC2530 芯片的基本功能，掌握 IAR 平台的项目创建与编译；学过 51 单片机的外部中断，了解“中断是‘暂停当前任务，处理紧急事件’的机制”，知道按键、LED 的基本电路连接。

知识缺口：不熟悉 CC2530 外部中断的引脚特性（如 P0 口部分引脚需配置为通用 IO）；对 CC2530 中断寄存器（如 IEN0、PICTL）的配置逻辑陌生；缺乏“CC2530 中断服务函数编写”的经验，与 51 单片机的中断函数格式差异大。

### 4. 教学目标

知识目标：能说出 CC2530 外部中断的常用引脚，掌握中断配置的 3 个核心步骤，理解中断服务函数的作用。

**能力目标：**能独立配置 CC2530 外部中断引脚，能编写中断服务函数，实现“按键中断控制 LED”功能。

**素养目标：**培养“分析中断故障”的排查能力（如中断不触发时，检查引脚配置、中断使能），树立“精准控制硬件”的工程意识。

## 5. 教学重点

CC2530 外部中断配置步骤（引脚方向设置→中断触发方式选择→中断使能）；

“按键中断控制 LED”实验的电路连接与代码编写。

## 6. 教学难点

CC2530 中断寄存器的配置（如通过 IEN0 寄存器使能总中断，通过 PICTL 寄存器设置触发沿）；

中断服务函数的编写规范（需清除中断标志位，避免重复触发）。

## 7. 课程思政

讲解“中断”的“优先级”特性（紧急事件优先处理），类比“生活中优先处理重要事情”（如消防员优先通行），引导学生树立“合理规划、分清主次”的时间管理意识；

实验中鼓励学生自主排查“中断不触发”的故障（如引脚接反、寄存器配置错误），培养“遇到问题不放弃，主动寻找解决方案”的坚韧品质。

## 8. 教学方法

理论教学：讲授法（结合 CC2530 引脚图与寄存器表）、对比分析法（CC2530 与 51 单片机中断配置的差异）；

实验教学：示范操作法（教师投屏编写代码，分步讲解寄存器配置）、分组实操法（2 人一组，一人连接电路，一人编写代码，协作完成实验）。

## 9. 教学程序

### (1) 导入 (15 分钟)

复习导入：提问“51 单片机中，按下按键如何让 LED 快速响应？”（答案：外部中断），追问“CC2530 作为 WSN 节点的核心芯片，如何通过中断实现按键快速控制？它的中断配置和 51 单片机有什么不同？”

实验预告：展示“按键中断控制 LED”的成品效果（按下按键 LED 亮，松开灭），激发学生动手兴趣。

## (2) 教授 (45 分钟)

理论讲解 (20 分钟) :

CC2530 外部中断引脚: 重点介绍 P0.0、P0.1 (常用低功耗引脚), 说明“需通过 P0SEL 寄存器将引脚配置为通用 IO (而非外设功能)”;

中断触发方式: ①上升沿 (引脚电平从低变高, 如按键松开); ②下降沿 (电平从高变低, 如按键按下); ③双边沿 (两种情况都触发), 通过 PICTL 寄存器设置;

配置步骤: ①引脚初始化 (设置 P0DIR 为输入, P0SEL 为通用 IO); ②中断使能 (IEN0 寄存器置 1, 使能总中断; IEN1 寄存器置 1, 使能 P0 口中断); ③编写中断服务函数 (函数名固定为“P0\_ISR”, 需清除 P0IFG 中断标志位)。

实验操作 (25 分钟) :

电路连接 (10 分钟): ①CC2530 开发板 P0.0 引脚接按键 (一端接引脚, 一端接地, 串联 1kΩ 上拉电阻); ②P1.0 引脚接 LED (串联 220Ω 限流电阻, 另一端接地);

代码编写 (10 分钟): 教师演示编写代码, 重点讲解: ①P0 口初始化 (P0SEL &= ~0x01; P0DIR &= ~0x01); ②中断配置 (IEN0 |= 0x80; IEN1 |= 0x01; PICTL |= 0x01); ③中断服务函数 (#pragma vector = P0INT\_VECTOR \_\_interrupt void P0\_ISR () { if (P0IFG & 0x01) { P1\_0 ^= 1; P0IFG &= ~0x01; } P0IF = 0; } ) ;

学生调试 (5 分钟): 烧录代码到 CC2530 开发板, 测试按键功能, 教师巡视, 帮助排查故障 (如“LED 不亮”可能是 P1.0 引脚配置错误, 需检查 P1DIR 是否设为输出)。

## (3) 总结 (10 分钟)

步骤梳理: CC2530 外部中断配置→代码编写→电路连接→调试;

故障排查清单: ①中断不触发: 检查中断使能寄存器 (IEN0、IEN1) 是否置 1; ②LED 不亮: 检查 P1 口方向 (P1DIR 是否设为输出)、限流电阻是否接对。

## (4) 作业

基础题: 修改实验代码, 实现“按下按键触发中断, LED 闪烁 3 次后熄灭” (在中断服务函数中添加延时循环);

拓展题: 尝试用 P0.1 引脚接第二个按键, 实现“两个按键分别控制两个 LED 的亮灭”, 提交代码与测试视频。

## (5) 反思

学生对“中断标志位清除”易遗忘 (未清除 P0IFG 会导致中断重复触发), 下次课需在代码中用“红色标注”提醒;

部分学生电路连接时忽略“上拉电阻”，导致按键无电平变化，需提前准备“带电阻的按键模块”，减少硬件故障。

## 第四周：CC2530 串口 UART0 发送字符串（9月29日 - 10月3日）

### 1. 教学课题和内容

**课题：**CC2530 串口 UART0 原理与字符串发送实操

**内容：**①CC2530 串口 UART0 的工作模式（异步通信）与核心参数（波特率、数据位、停止位、校验位）；②UART0 初始化步骤（引脚功能配置、波特率设置、通信参数配置）；③字符串发送原理（逐字符发送，结合循环语句）；④实验：基于 UART0 的字符串发送（通过串口将“WSN Node 1”发送到电脑，用串口助手接收并显示）。

### 2. 教学课时

总学时 4 学时（理论 2 学时 + 实验 2 学时）

### 3. 学生已有知识分析

**已有基础：**学生已掌握 CC2530 引脚配置（如 GPIO 初始化），熟悉 IAR 平台项目创建与编译；了解串口通信基本概念（如波特率表示数据传输速率），部分学生用串口助手接收过 51 单片机发送的数据。

**知识缺口：**不熟悉 CC2530 UART0 的硬件结构（如 U0BAUD、U0GCR 等寄存器）；对“波特率与寄存器值的对应关系”陌生，难以自主计算配置值；缺乏“字符串逐字符发送”的代码编写经验，不清楚串口中断与查询发送的差异。

### 4. 教学目标

**知识目标：**理解 CC2530 UART0 的异步通信原理，掌握 UART0 初始化的 3 个核心步骤，明确字符串发送的实现逻辑。

**能力目标：**能独立配置 UART0 的波特率（如 9600bps），能编写字符串发送代码，用串口助手接收并验证数据。

**素养目标：**培养“硬件配置与软件编程结合”的工程思维，通过串口数据校验培养“追求数据准确性”的严谨态度。

### 5. 教学重点

CC2530 UART0 初始化（引脚复用为 UART 功能、波特率寄存器配置、通信参数设置）；字符串发送代码编写（循环遍历字符串，调用发送函数逐字符发送）。

### 6. 教学难点

UART0 波特率的寄存器配置（根据目标波特率计算 U0BAUD 与 U0GCR 的值，如 9600bps 对应 U0BAUD=216，U0GCR=11）；

串口通信故障排查（如波特率不匹配导致接收乱码、引脚接反导致无数据）。

## 7. 课程思政

结合“串口数据传输的准确性”，讲解物联网数据采集需“真实、精准”（如环境监测数据误差会影响决策），培养学生的诚信品质与数据责任意识；

展示工业场景中“串口用于设备调试”的案例（如通过串口读取 WSN 节点运行状态），说明“调试工具是排查问题的关键”，引导学生重视基础工具的使用。

## 8. 教学方法

理论教学：讲授法（结合 UART0 寄存器框图）、公式推导法（演示波特率与寄存器值的计算）；

实验教学：演示教学法（教师投屏编写初始化代码与发送代码）、分组实操法（2 人一组，一人配置串口助手，一人编写代码，协作验证数据）。

## 9. 教学程序

### (1) 导入 (20 分钟)

问题导入：“上节课我们用中断控制 LED，如何将节点状态（如‘LED 已点亮’）发送到电脑？为什么工业设备常用串口调试？”引出 CC2530 串口 UART0；

实物演示：用串口助手展示“电脑接收单片机发送的‘Hello’”，提问“要实现 CC2530 向电脑发字符串，需要配置哪些参数？”

### (2) 教授 (100 分钟)

理论讲解（40 分钟）：

UART0 工作原理：异步通信（无需时钟线，通过波特率同步），核心参数：①波特率（常用 9600bps、115200bps）；②数据位（8 位，默认）；③停止位（1 位，默认）；④校验位（无校验，简化配置）；

初始化步骤：①引脚配置（将 P0.2 设为 UART0\_TX，P0.3 设为 UART0\_RX，通过 P0SEL 寄存器实现引脚复用）；②波特率设置（U0GCR 寄存器控制波特率指数，U0BAUD 控制波特率小数部分，公式：波特率 = (系统时钟 / (16 × (2<sup>U0GCR</sup> × (U0BAUD + 1))) )，系统时钟默认 32MHz）；③通信参数配置（U0CSR 寄存器设为 8 位数据、1 位停止位、无校验，U0UCR 寄存器使能发送）；

字符串发送逻辑：定义字符串数组（如 char str [] = "WSN Node 1"），通过 for 循环遍历数组，调用发送函数（如 void UART0\_SendChar (char c) { while (!(U0CSR & 0x02)); U0DBUF = c; }）逐字符发送。

实验操作（60 分钟）：

电路连接（10 分钟）：①CC2530 开发板 P0.2 (TX) 接 USB 转 TTL 模块的 RX；②P0.3 (RX) 接 USB 转 TTL 模块的 TX；③USB 转 TTL 模块接电脑 USB 口（供电 + 通信）；

代码编写（30 分钟）：①UART0 初始化函数（void UART0\_Init (void) { P0SEL |= 0x0C; U0GCR = 11; U0BAUD = 216; U0CSR = 0x80; U0UCR = 0x80; }）；②字符串发送函数（void UART0\_SendStr (char \*str) { while (\*str != '\0') { UART0\_SendChar (\*str); str++; } }）；③主函数（调用初始化函数，延时 1 秒后发送字符串）；

串口助手配置与验证（20 分钟）：①打开串口助手（如 SSCOM），选择对应 COM 口，设置波特率 9600bps、8 位数据、1 位停止位、无校验；②烧录代码到 CC2530，观察串口助手是否显示“WSN Node 1”，排查故障（如乱码 → 检查波特率，无数据 → 检查引脚连接）。

### (3) 总结 (15 分钟)

流程梳理：UART0 参数确定（波特率等）→寄存器配置→代码编写→串口助手验证；  
故障排查清单：①乱码：波特率不匹配、校验位错误；②无数据：引脚接反（TX-RX 交叉）、UART0 初始化函数未调用。

## (4) 作业

基础题：修改代码，实现每隔 2 秒发送一次“Temperature: 25°C”；

拓展题：查询 CC2530 UART0 支持的其他波特率（如 115200bps），计算对应的 U0BAUD 与 U0GCR 值，编写代码并验证发送。

## (5) 反思

学生易在“波特率寄存器计算”环节出错，下次课可提供“波特率 - 寄存器值对应表”，避免重复计算；

部分学生对“引脚复用”理解困难，可通过“引脚功能切换动画”（如 P0.2 从 GPIO 切换为 UART\_TX）强化认知。

# 第五周：温湿度传感器 DHT11 的使用（10 月 6 日 - 10 月 10 日）

## 1. 教学课题和内容

课题：DHT11 传感器原理与 CC2530 数据采集

内容：①DHT11 的工作模式（单总线通信）与数据格式（40 位数据，含湿度整数、湿度小数、温度整数、温度小数、校验和）；②DHT11 数据采集时序（起始信号、存在信号、数据位读取）；③CC2530 与 DHT11 的硬件连接与代码实现；④实验：DHT11 温湿度采集（采集环境温湿度，通过串口将“Humidity: 60%, Temperature: 26°C”发送到电脑）。

## 2. 教学课时

总学时 2 学时（理论 1 学时 + 实验 1 学时）

## 3. 学生已有知识分析

已有基础：学生已掌握 CC2530 GPIO 操作（如引脚输入 / 输出配置），熟悉串口 UART0 发送数据；了解“单总线”是“一根线实现数据双向传输”，对传感器“采集 - 处理 - 传输”的流程有初步认知。

知识缺口：不熟悉 DHT11 的严格时序要求（如起始信号低电平需持续 18ms 以上）；对“数据位（0/1）的时序区分”（低电平后高电平持续时间：0→26-28μs，1→70μs）陌生；缺乏“校验和验证数据有效性”的代码编写经验。

## 4. 教学目标

知识目标：理解 DHT11 单总线通信原理，掌握数据采集的 3 个核心时序（起始、存在、数据读取），明确校验和的作用（验证数据是否正确）。

能力目标：能独立连接 CC2530 与 DHT11，能编写数据采集与校验代码，通过串口正确输出温湿度。

**素养目标：**培养“精准控制时序”的工程意识，通过数据校验培养“质疑数据、验证真实性”的科学态度。

## 5. 教学重点

DHT11 数据采集时序（起始信号的低电平时长、存在信号的检测、数据位 0/1 的时序区分）；  
CC2530 与 DHT11 的硬件连接（单总线引脚选择、上拉电阻作用）。

## 6. 教学难点

DHT11 微秒级时序的精准控制（需编写延时函数，避免时序偏差导致采集失败）；  
校验和计算与数据有效性判断（校验和 = 湿度整数 + 湿度小数 + 温度整数 + 温度小数，若不相等则数据无效）。

## 7. 课程思政

展示 DHT11 在“冷链物流温湿度监测”中的应用（实时采集运输过程中食品的温湿度，超阈值报警），说明技术对“食品安全保障”的重要性，培养学生的社会责任感；  
强调“时序偏差会导致采集失败”，类比“生活中‘按时赴约’的重要性”，引导学生树立“守时、精准”的行为习惯。

## 8. 教学方法

理论教学：讲授法（结合 DHT11 时序波形图）、动画演示法（播放“起始 - 存在 - 数据读取”完整时序动画）；  
实验教学：示范操作法（教师演示延时函数编写与时序调试）、分组实操法（3 人一组，一人连接电路，一人编写代码，一人验证数据）。

## 9. 教学程序

### (1) 导入 (15 分钟)

场景导入：“智慧大棚需要实时监测温湿度，若湿度低于 50% 需自动浇水，温度高于 30℃ 需开窗通风，核心是如何精准采集温湿度？”引出 DHT11；

实物展示：拿出 DHT11 传感器，提问“它只有 3 个引脚（VCC、GND、DATA），如何通过一根 DATA 线传输温湿度数据？”

### (2) 教授 (45 分钟)

理论讲解（20 分钟）：

DHT11 工作原理：单总线通信，DATA 线既传命令又传数据，供电 3.3-5V，采集周期 $\geq 1$  秒；  
数据格式：40 位数据，按“湿度整数（8 位） $\rightarrow$ 湿度小数（8 位，通常为 0） $\rightarrow$ 温度整数（8 位） $\rightarrow$ 温度小数（8 位，通常为 0） $\rightarrow$ 校验和（8 位）”顺序传输；

采集时序：①起始信号：CC2530 拉低 DATA 线 18ms 以上，再拉高 20-40 $\mu$ s；②存在信号：DHT11 拉低 DATA 线 80 $\mu$ s，再拉高 80 $\mu$ s，表明已准备好数据；③数据读取：DHT11 拉低 DATA 线 50 $\mu$ s 后，拉高时长决定数据（26-28 $\mu$ s $\rightarrow$ 0，70 $\mu$ s $\rightarrow$ 1），CC2530 需在拉高期间读取电平。

实验操作（25 分钟）：

电路连接（8分钟）：①DHT11 VCC 接 CC2530 3.3V；②GND 接 GND；③DATA 接 CC2530 P0.4；④DATA 线串联 10kΩ 上拉电阻（保证空闲时为高电平）；

代码编写（12分钟）：①微秒级延时函数（`void Delay_us (uint16 t){ while (t--); }`，需通过示波器校准）；②DHT11 初始化函数（`void DHT11_Init (void){ P0DIR |= 0x10; P0_4=1; Delay_ms (100); }`）；③数据采集函数（`uint8 DHT11_ReadData (uint8 buf){ /实现起始信号、存在信号检测、数据读取与校验 */ }`）；④主函数（循环采集温湿度，通过串口发送）；

数据验证（5分钟）：烧录代码，用串口助手查看数据，若显示“Humidity: 60%, Temperature: 26°C”则成功，若数据异常（如校验和错误），检查时序与上拉电阻。

### (3) 总结（10分钟）

时序梳理：起始信号（CC2530 主动）→存在信号（DHT11 响应）→数据读取（双向交互）；

核心要点：上拉电阻不可少（防止 DATA 线电平不稳定），延时函数需精准（误差 $\leq 10\mu\text{s}$ ），校验和必验证（排除错误数据）。

### (4) 作业

基础题：修改代码，当湿度  $< 50\%$  时，串口额外发送“Need Water!”，温度  $> 30^\circ\text{C}$  时发送“Need Cool!”；

拓展题：尝试用 P0.5 引脚连接第二个 DHT11，实现双传感器同时采集，提交串口显示截图。

### (5) 反思

学生易在“微秒级延时校准”上出现问题，下次课可提供“已校准的延时函数库”，减少调试时间；部分学生对“数据位读取逻辑”理解困难，可通过“时序波形与代码行对应标注”（如“DHT11 拉低  $50\mu\text{s}$ →代码中 `Delay_us (50);`”）帮助理解。

## 第六周：PWM 调光与基础篇总结（10月13日 - 10月17日）

### 1. 教学课题和内容

课题：CC2530 PWM 调光原理与基础内容总结

内容：①PWM（脉冲宽度调制）原理（通过改变高电平占空比调节输出平均电压）；②CC2530 定时器 1 生成 PWM 的配置（定时器模式、比较模式、占空比设置）；③PWM 调光逻辑（占空比越大，LED 越亮）；④基础篇总结（CC2530 GPIO、中断、串口、传感器、PWM 的知识关联）；⑤实验：基于定时器 1 的 PWM 调光（通过按键调节占空比，实现 LED 从暗到亮、从亮到暗渐变）。

### 2. 教学课时

总学时 4 学时（理论 2 学时 + 实验 2 学时）

### 3. 学生已有知识分析

已有基础：学生已掌握 CC2530 定时器基础（如定时器中断），熟悉 GPIO 输出控制（点亮 LED）；了解“占空比”是“高电平时间占周期的比例”，对“PWM 用于调光、调速”有初步认知。

**知识缺口：**不熟悉 CC2530 定时器 1 的比较模式（生成 PWM 的核心模式）；对“定时器周期与 PWM 频率的关系”（频率 = 1 / 周期）陌生；缺乏“PWM 占空比动态调节”的代码编写经验，难以理解定时器中断与 PWM 的协同。

## 4. 教学目标

**知识目标：**理解 PWM 调光原理，掌握 CC2530 定时器 1 生成 PWM 的配置步骤，能梳理基础篇核心知识的逻辑关联。

**能力目标：**能独立配置定时器 1 生成 PWM，能通过按键动态调节占空比实现 LED 调光，能绘制基础篇知识思维导图。

**素养目标：**培养“从‘控制开关’到‘精准调节’的进阶思维”，通过基础总结培养“系统整合知识”的学习习惯。

## 5. 教学重点

CC2530 定时器 1 生成 PWM 的配置（定时器周期设置、比较值与占空比的关系）；  
基础篇知识关联（GPIO→中断→串口→传感器→PWM 的应用场景串联）。

## 6. 教学难点

定时器 1 比较模式的理解（通过比较值与定时器计数值的匹配，控制引脚电平翻转）；  
PWM 占空比动态调节（在定时器中断中修改比较值，避免 PWM 波形紊乱）。

## 7. 课程思政

结合“PWM 调光节能”（相比传统电阻调光，PWM 调光功耗低 30% 以上），讲解低碳理念在电子设计中的应用，引导学生设计“低功耗 WSN 节点”，培养环保意识；  
基础篇总结时，回顾“从硬件配置到软件实现”的学习历程，强调“循序渐进、夯实基础”的重要性，鼓励学生树立“脚踏实地、逐步提升”的学习态度。

## 8. 教学方法

理论教学：讲授法（结合 PWM 波形图与定时器框图）、思维导图法（梳理基础篇知识）；  
实验教学：示范操作法（教师演示定时器配置与占空比调节代码）、分组合作法（2 人一组，一人负责 PWM 配置，一人负责按键中断调节占空比）。

## 9. 教学程序

### (1) 导入 (20 分钟)

现象导入：展示“LED 从暗到亮渐变”的效果，提问“为什么 LED 亮度能连续变化？不是只有‘亮’和‘灭’两种状态吗？”引出 PWM 调光；

回顾预告：“本周除了学习 PWM，还要总结前 5 周的基础内容，形成完整的知识体系，为后续无线通信学习打基础。”

### (2) 教授 (100 分钟)

理论讲解（40 分钟）：

PWM 原理：①周期  $T$ （高电平 + 低电平时间），频率  $f=1/T$ （常用 1kHz，人眼无闪烁）；②占空比  $D = \text{高电平时间} / T \times 100\%$ ， $D$  越大，LED 平均电压越高，越亮；

定时器 1 配置（生成 PWM）：①定时器模式：设为模模式（计数到 T1CC0 值后重置），T1CC0 决定周期（如 T1CC0=3200，系统时钟 32MHz，周期 =  $3200/(32\text{MHz}/8)=800\mu\text{s}$ ，频率 = 1250Hz）；②比较模式：T1CC1 设为比较值，当定时器计数值 = T1CC1 时，引脚电平翻转，占空比 =  $T1CC1/T1CC0 \times 100\%$ （如 T1CC1=1600， $D=50\%$ ）；③引脚配置：将 P1.0 设为定时器 1 输出引脚（通过 P1SEL 寄存器）；

基础篇总结：①核心硬件：CC2530 芯片（GPIO、定时器、串口、中断）；②外设交互：DHT11（传感器）、LED / 按键（执行器）；③应用逻辑：采集（传感器）→处理（CC2530）→传输（串口）→控制（LED/PWM）。

实验操作（60 分钟）：

电路连接（10 分钟）：①LED 串联 220Ω 限流电阻接 CC2530 P1.0；②按键（带 10kΩ 上拉电阻）接 P0.0（用于调节占空比，按下一次占空比 + 10%，满 100% 后重置为 10%）；

代码编写（35 分钟）：①定时器 1 初始化函数（`void Timer1_PWM_Init(void){ T1CTL=0x02; T1CC0=3200; T1CCTL1=0x24; P1SEL |= 0x01; }`）；②按键中断函数（检测到按键按下，修改 T1CC1 值，如 `T1CC1 += 320; if (T1CC1>3200) T1CC1=3200;`）；③主函数（初始化定时器、按键中断，循环等待）；

功能验证（15 分钟）：烧录代码，按下按键观察 LED 亮度变化，用示波器测量 P1.0 引脚的 PWM 波形，验证占空比是否随按键变化（如 10%→20%→...→100%）。

### (3) 总结 (15 分钟)

PWM 逻辑梳理：定时器周期（决定频率）→比较值（决定占空比）→引脚输出（控制 LED 亮度）；

基础篇知识导图：以“CC2530”为核心，分支为“GPIO（控制）、中断（响应）、串口（传输）、传感器（采集）、PWM（调节）”，标注各模块的关联（如传感器采集数据→串口传输→PWM 根据数据调节）。

### (4) 作业

基础题：修改代码，实现“占空比从 10% 到 90%，每 2 秒增加 10%，再从 90% 减到 10%”的自动渐变；

拓展题：绘制基础篇知识思维导图，标注每个模块的核心知识点与应用场景，提交图片。

### (5) 反思

学生易在“定时器周期与频率的计算”上出错，下次课可提供“频率 - 周期 - CC0 值对应表”（如 1kHz 对应 CC0=3200）；

部分学生对“比较模式生成 PWM”的原理理解困难，可通过“定时器计数与比较值匹配的动画”（如计数值从 0 到 3200，到 1600 时引脚翻转）强化认知。

## 第七周：协议栈工作流程和无线收发控制 LED (10 月 20 日 - 10 月 24 日)

### 1. 教学课题和内容

课题：Zstack 协议栈工作流程与无线收发控制 LED

**内容:** ①Zstack 协议栈的核心架构 (OSAL 操作系统抽象层、网络层、应用层的交互); ②协议栈初始化流程 (系统初始化→OSAL 初始化→应用层初始化); ③无线数据收发原理 (协调器创建网络, 终端节点加入网络, 基于端点号、簇 ID 通信); ④实验: 无线收发控制 LED (终端节点发送“LED\_ON”, 协调器接收后点亮 LED; 发送“LED\_OFF”, 协调器熄灭 LED)。

## 2. 教学课时

总学时 2 学时 (理论 1 学时 + 实验 1 学时)

## 3. 学生已有知识分析

**已有基础:** 学生已理解 Zigbee 协议栈分层 (物理层到应用层), 熟悉 CC2530 硬件操作 (GPIO、串口); 了解“协调器创建网络、终端节点加入网络”的基本概念, 对“无线通信需地址匹配”有初步认知。

**知识缺口:** 不熟悉 Zstack 协议栈的 OSAL 任务调度机制 (多任务并行处理); 对“端点号、簇 ID”的作用 (标识通信对象与数据类型) 陌生; 缺乏“协议栈下无线收发代码编写”的经验, 不清楚数据帧的封装与解析。

## 4. 教学目标

**知识目标:** 理解 Zstack 协议栈的 OSAL 任务调度与初始化流程, 掌握无线收发的核心参数 (端点号、簇 ID、网络地址), 明确“发送 - 接收 - 控制”的逻辑链。

**能力目标:** 能配置协调器与终端节点的协议栈参数, 能编写无线收发代码控制 LED, 能验证网络组建与数据传输。

**素养目标:** 培养“基于协议栈开发”的工程思维, 通过无线协作控制培养“节点间协同”的团队意识。

## 5. 教学重点

Zstack 协议栈初始化流程 (系统→OSAL→应用层的调用顺序);  
无线收发代码编写 (数据帧封装、发送函数调用、接收回调函数处理)。

## 6. 教学难点

OSAL 任务调度机制的理解 (任务优先级、事件触发, 如 SYS\_EVENT\_MSG 事件);  
无线数据帧的解析 (从接收数据中提取有效控制指令, 如“LED\_ON”)。

## 7. 课程思政

讲解“Zstack 协议栈的协作性” (协调器与终端节点分工: 协调器管理网络, 终端节点采集 / 控制), 类比团队项目中“分工明确、协同高效”的重要性, 引导学生树立团队协作意识;  
展示“WSN 无线控制在智能家居中的应用” (如手机通过终端节点控制客厅 LED), 说明“技术简化生活、提升体验”, 激发学生对物联网行业的热爱。

## 8. 教学方法

理论教学: 讲授法 (结合 Zstack 协议栈架构图)、流程图法 (绘制协议栈初始化与无线收发流程);

实验教学：示范操作法（教师演示协议栈工程修改与代码编写）、分组实操法（2组对比，一组做协调器，一组做终端节点，验证无线控制）。

## 9. 教学程序

### (1) 导入 (15 分钟)

问题导入：“之前我们的通信都需要有线（如串口线），WSN 节点如何实现无线通信？协调器和终端节点如何‘认识’对方并传输数据？”引出 Zstack 协议栈；

实物演示：展示“终端节点按下按键，协调器 LED 点亮”的效果，提问“这个过程中协议栈做了哪些工作？”

### (2) 教授 (45 分钟)

理论讲解（20 分钟）：

Zstack 协议栈架构：①OSAL 层（管理任务，如应用任务、网络任务）；②网络层（负责组网，如协调器创建网络、终端节点加入）；③应用层（定义用户功能，如无线收发、LED 控制）；

初始化流程：①系统初始化（硬件初始化，如时钟、GPIO）；②OSAL 初始化（创建任务，分配优先级，应用任务优先级低于系统任务）；③应用层初始化（注册端点号、簇 ID，初始化外设如 LED）；

无线收发核心参数：①端点号（如 0x01，标识应用模块）；②簇 ID（如 0x0001，标识数据类型，如控制指令）；③网络地址（协调器默认 0x0000，终端节点加入后分配唯一地址）；

收发流程：①发送：终端节点调用 AF\_DataRequest () 函数，封装数据帧（含目标地址、端点号、簇 ID）；②接收：协调器注册 AF\_DataConfirm () 回调函数，接收数据后解析指令，控制 LED。

实验操作（25 分钟）：

工程准备（5 分钟）：①打开 Zstack 协议栈的 GenericApp 工程（协议栈自带示例，适配 CC2530）；②协调器工程：在 f8wConfig.cfg 中设 DEVICE\_TYPE=COORDINATOR；③终端节点工程：设 DEVICE\_TYPE=END\_DEVICE；

代码修改（15 分钟）：①协调器（GenericApp.c）：在 GenericApp\_ProcessEvent () 函数中，接收数据时解析（if (strstr ((char\*) pkt->cmd.Data, "LED\_ON")) P1\_0=0; else if (strstr ((char\*) pkt->cmd.Data, "LED\_OFF")) P1\_0=1;); ②终端节点（GenericApp.c）：按键中断中调用发送函数（ AF\_DataRequest (&GenericApp\_DstAddr, &GenericApp\_EndPointDesc, 0x0001, strlen ("LED\_ON")+1, (uint8\*)"LED\_ON", &GenericApp\_TransID, AF\_DISCV\_ROUTE, AF\_DEFAULT\_RADIUS) );

功能验证（5 分钟）：①烧录协调器程序，上电后创建网络（LED 闪烁 3 次表示组网成功）；②烧录终端节点程序，上电后加入网络（LED 常亮表示加入成功）；③按下终端节点按键，观察协调器 LED 是否点亮 / 熄灭。

### (3) 总结 (10 分钟)

协议栈流程梳理：初始化（系统→OSAL→应用）→组网（协调器创建→终端加入）→收发（终端发送→协调器接收→控制）；

核心要点：端点号与簇 ID 需匹配（否则无法通信），网络地址需正确（终端节点需知道协调器地址 0x0000），回调函数需注册（否则无法接收数据）。

### (4) 作业

基础题：修改终端节点代码，实现“按下按键发送‘LED\_TOGGLE’，协调器接收后翻转 LED 状态”；

拓展题：尝试在协调器中添加串口发送功能，接收终端节点指令后，通过串口将“Received: LED\_ON”发送到电脑，提交测试视频。

## (5) 反思

学生易在“协议栈工程配置”上出错（如 DEVICE\_TYPE 设错导致无法组网），下次课可提供“协调器 / 终端节点配置模板”，减少配置错误；

部分学生对“AF\_DataRequest () 函数参数”理解困难，可通过“参数注释表”（如目标地址、端点号、数据长度的含义）帮助记忆。

## 第八周：无线数据传输（10月27日 - 10月31日）

### 1. 教学课题和内容

**课题：**WSN 多节点无线数据传输与组网优化

**内容：**①多节点组网结构（协调器 + 多个终端节点，星型拓扑）；②无线数据帧结构（帧头、目标地址、源地址、端点号、簇 ID、数据、校验和）；③数据传输优化（重传机制、信道选择，避免干扰）；④实验：多节点温湿度数据传输（3个终端节点采集温湿度，无线发送到协调器，协调器通过串口显示“Node 1: H=60%, T=25°C; Node 2: H=58%, T=26°C; Node 3: H=62%, T=24°C”）。

### 2. 教学课时

总学时 4 学时（理论 2 学时 + 实验 2 学时）

### 3. 学生已有知识分析

**已有基础：**学生已掌握 Zstack 协议栈下的点对点无线收发（终端→协调器），熟悉 DHT11 温湿度采集与串口发送；了解星型拓扑“协调器为中心，终端节点围绕”的结构，对“多节点需区分地址”有初步认知。

**知识缺口：**不熟悉多节点数据的区分方法（如通过源地址识别不同终端节点）；对“无线数据重传机制”（数据丢失后自动重新发送）陌生；缺乏“多节点同时传输的冲突避免”经验，易出现数据覆盖。

### 4. 教学目标

**知识目标：**理解 WSN 星型拓扑的多节点组网原理，掌握无线数据帧的核心字段（源地址、数据），明确数据传输优化方法（重传、信道选择）。

**能力目标：**能实现 3 个终端节点向协调器传输温湿度数据，能通过源地址区分不同节点数据，能优化传输参数减少数据丢失。

**素养目标：**培养“多节点系统设计”的全局思维，通过数据传输优化培养“追求高效、可靠”的工程素养。

### 5. 教学重点

多节点数据区分（通过 AF\_INCOMING\_MSG\_CMD 事件中的源地址字段识别节点）；  
无线数据传输优化（设置重传次数、选择空闲信道）。

## 6. 教学难点

多节点同时传输的冲突处理（通过设置不同发送周期，避免数据同时到达协调器）；  
数据帧解析与多节点数据格式化显示（协调器接收多个节点数据后，按节点编号排序显示）。

## 7. 课程思政

展示“无线数据传输在地震救援中的应用”（多个 WSN 节点部署在废墟中，实时传输幸存者位置与环境数据到指挥中心），说明技术对“生命救援”的重要性，培养学生的使命感与责任感；  
强调“多节点数据传输的可靠性”（数据丢失可能导致救援决策失误），类比“生活中‘重要信息需确认传达’”，引导学生树立“重视细节、保障可靠”的意识。

## 8. 教学方法

理论教学：讲授法（结合多节点组网拓扑图与数据帧结构）、案例分析法（分析多节点数据冲突的解决案例）；

实验教学：分组合作法（3组各做1个终端节点，1组做协调器，共同完成多节点传输）、故障排查法（针对数据丢失、冲突问题，引导学生优化参数）。

## 9. 教学程序

### (1) 导入 (20 分钟)

场景导入：“智慧农业中，需要在大棚不同位置部署 10 个温湿度节点，如何让所有节点的数据都能准确传输到中控协调器？多个节点同时发送数据会出现什么问题？”引出多节点无线传输；

问题讨论：“之前点对点传输只需知道目标地址，多节点传输还需要什么信息来区分不同节点的数据？”

### (2) 教授 (100 分钟)

理论讲解（40 分钟）：

多节点组网结构：星型拓扑，协调器（0x0000）为中心，终端节点（0x0001、0x0002、0x0003）分散部署，各终端节点直接与协调器通信，不相互通信；

无线数据帧结构：①帧头（标识帧类型，如数据帧）；②目标地址（协调器地址 0x0000）；③源地址（终端节点地址，如 0x0001）；④端点号（0x01，统一应用端点）；⑤簇 ID（0x0002，温湿度数据簇）；⑥数据（如“60,25”表示湿度 60%、温度 25°C）；⑦校验和（验证数据完整性）；

传输优化方法：①重传机制：在 `AF_DataRequest()` 函数中设置重传次数（如 `AF_DEFAULT_RADIUS=3`，最多重传 3 次）；②信道选择：通过协议栈工具（如 Z-Tool）扫描空闲信道（如 2.4GHz 频段的 11-26 信道，避开 WiFi 常用的 1、6、11 信道）；③冲突避免：为不同终端节点设置不同发送周期（如节点 1 每 3 秒发送，节点 2 每 4 秒，节点 3 每 5 秒）。

实验操作（60 分钟）：

节点配置（15 分钟）：①协调器：GenericApp 工程设 `DEVICE_TYPE=COORDINATOR`，初始化串口与 LED，注册数据接收回调函数；②终端节点 1-3：分别设 `DEVICE_TYPE=END_DEVICE`，修改发送周期（`node1:3s`，`node2:4s`，`node3:5s`），DHT11 接各自的 P0.4 引脚；

代码编写（30 分钟）：①终端节点：采集温湿度后，将数据格式化为“节点号，H,T”（如“1,60,25”），调用 `AF_DataRequest()` 发送；②协调器：在 `GenericApp_ProcessEvent()` 中，接收数据时提取源地址（`pkt->srcAddr.addr.shortAddr`）与数据，按节点号排序，通过串口格式化显示（如“Node 1: H=60%, T=25°C; Node 2: H=58%, T=26°C;”）；

功能验证与优化（15分钟）：①各节点上电，协调器确认所有节点加入网络（串口显示“Node 0x0001 Joined”）；②观察串口显示，若数据完整无丢失则成功；若出现数据覆盖，调整终端节点发送周期（如增大周期差）；若数据丢失，增加重传次数（如 AF\_DEFAULT\_RADIUS=5）。

### (3) 总结（15分钟）

多节点传输逻辑：节点区分（源地址）→冲突避免（不同周期）→可靠传输（重传+信道选择）→格式化显示（协调器）；

优化要点：发送周期差 $\geq 1$ 秒（避免同时发送），重传次数 $\geq 3$ 次（应对短暂干扰），信道选择空闲频段（减少 WiFi 干扰）。

### (4) 作业

基础题：增加第4个终端节点，设置发送周期6秒，修改协调器代码实现4个节点数据的格式化显示；

拓展题：在协调器中添加“数据丢失报警”功能，若10秒内未收到某个节点数据，串口发送“Node X Data Lost!”，提交测试截图。

### (5) 反思

学生易在“源地址提取与节点号匹配”上出错（如将源地址 0x0001 直接当作节点号 1），下次课可在代码中添加“源地址→节点号映射表”（如 `addrMap [0x0001]=1`，`addrMap [0x0002]=2`）；

部分学生对“信道选择”操作不熟悉，下次课可演示 Z-Tool 扫描信道的步骤，帮助学生找到空闲信道。

## 第九周：组播组网 - 多终端控制协调器 LED（11月3日 - 11月7日）

### 1. 教学课题和内容

课题：WSN 组播组网原理与多终端控制实操

内容：①组播通信的定义（一对多通信，协调器为接收方，多个终端为发送方）与组播地址（如 0x0001，标识组播组）；②Zstack 协议栈组播配置（组播组创建、终端节点加入组播组）；③多终端控制逻辑（终端节点发送组播指令，协调器接收后执行 LED 控制）；④实验：多终端组播控制 LED（3个终端节点均可发送“LED\_ON”/“LED\_OFF”组播指令，协调器接收任意终端指令后控制 LED，实现“多控一”）。

### 2. 教学课时

总学时 2 学时（理论 1 学时 + 实验 1 学时）

### 3. 学生已有知识分析

已有基础：学生已掌握 WSN 星型拓扑的点对点传输（多终端→协调器），熟悉 Zstack 协议栈的 `AF_DataRequest()` 函数；了解“地址用于标识通信对象”，对“一对多通信比点对点更高效”有初步认知。

**知识缺口：**不熟悉组播地址的分配与作用（区别于单播地址，组播地址对应一组节点）；对“Zstack 组播组创建与加入的 API 函数”（如 `NLME_JoinMulticastGroup()`）陌生；缺乏“多终端发送组播指令的冲突处理”经验，易出现指令覆盖。

## 4. 教学目标

**知识目标：**理解组播通信的“一对多”特性，掌握 Zstack 组播配置的 2 个核心步骤（创建组播组、加入组播组），明确多终端组播控制的逻辑。

**能力目标：**能创建组播组并让终端节点加入，能编写组播指令发送与接收代码，实现多终端控制协调器 LED。

**素养目标：**培养“高效通信设计”的思维，通过“多控一”场景培养“协同控制”的团队意识。

## 5. 教学重点

Zstack 组播配置（协调器创建组播组，终端节点调用 `NLME_JoinMulticastGroup()` 加入）；组播指令的发送与接收（目标地址设为组播地址，协调器接收组播数据）。

## 6. 教学难点

组播地址与单播地址的区分（组播地址通常为 `0x0001-0xFFFF` 中的特定值，不与终端单播地址重复）；

多终端同时发送组播指令的冲突处理（通过指令优先级或时间戳，避免协调器执行错误指令）。

## 7. 课程思政

结合“组播组网的高效性”（一次发送，多个节点接收，减少通信开销），类比“团队会议中‘一人传达，多人听取’的高效协作模式”，引导学生树立“高效沟通、资源共享”的意识；

展示“组播在智慧楼宇照明控制中的应用”（多个办公室的控制面板通过组播指令，共同控制大厅主灯），说明技术对“便捷管理、节能降耗”的作用，培养学生的应用思维。

## 8. 教学方法

理论教学：讲授法（结合组播组网拓扑图）、对比分析法（组播与单播的通信效率、地址类型差异）；

实验教学：示范操作法（教师演示组播组创建与加入代码）、分组实操法（3 组做终端节点，1 组做协调器，验证多终端组播控制）。

## 9. 教学程序

### (1) 导入 (15 分钟)

问题导入：“之前 3 个终端节点控制协调器 LED，需要每个终端都向协调器单播发送指令，有没有更高效的方式让所有终端‘一次约定，共同控制’？”引出组播通信；

对比导入：展示“单播（3 次发送）vs 组播（1 次发送，3 个终端加入组播组）”的通信流程，提问“组播为什么更高效？需要什么特殊地址？”

### (2) 教授 (45 分钟)

理论讲解（20分钟）：

组播通信原理：①定义：一对多通信，发送方（终端节点）向组播地址发送数据，所有加入该组播组的接收方（协调器）均可接收；②优势：减少通信次数（单播需N次，组播需1次），降低网络开销；③组播地址：Zstack中常用0x0001作为组播组地址，需确保不与终端单播地址（如0x0001-0x0003）重复（可设组播地址为0x1000）；

Zstack组播配置步骤：①协调器创建组播组：调用NLME\_CreateMulticastGroup()函数，设置组播地址（如0x1000）、端点号（0x01）；②终端节点加入组播组：调用NLME\_JoinMulticastGroup()函数，传入组播地址（0x1000）与端点号（0x01），返回成功则加入；

多终端控制逻辑：①终端节点：发送指令时，目标地址设为组播地址（0x1000），数据为“LED\_ON”/“LED\_OFF”；②协调器：加入组播组后，接收所有发往0x1000的组播数据，解析指令并控制LED，无需区分具体终端节点。

实验操作（25分钟）：

代码修改（15分钟）：①协调器（GenericApp.c）：在GenericApp\_Init()中创建组播组（NLME\_CreateMulticastGroup(0x1000, &GenericApp\_EndPointDesc);），接收数据时直接解析指令；②终端节点1-3（GenericApp.c）：在GenericApp\_Init()中加入组播组（NLME\_JoinMulticastGroup(0x1000, &GenericApp\_EndPointDesc);），按键中断中发送组播指令（AF\_DataRequest()的目标地址设为0x1000）；

电路与烧录（5分钟）：①协调器LED接P1.0；②终端节点各接1个按键（控制发送“LED\_ON”/“LED\_OFF”）；③分别烧录协调器与终端节点程序；

功能验证（5分钟）：①协调器上电，创建组播组（LED闪烁2次表示成功）；②终端节点上电，加入组播组（LED常亮表示成功）；③按下任意终端节点按键，观察协调器LED是否按指令动作（如节点1按“ON”则亮，节点2按“OFF”则灭）。

### (3) 总结（10分钟）

组播流程梳理：协调器创建组播组→终端加入组播组→终端发送组播指令→协调器接收执行；

核心对比：单播（目标地址 = 终端单播地址，一对一）vs 组播（目标地址 = 组播地址，一对多），组播更适合“多发送方、同一接收方”的场景。

### (4) 作业

基础题：修改代码，实现“终端节点发送‘LED\_TOGGLE’组播指令，协调器接收后翻转LED状态”，3个终端均可触发；

拓展题：在协调器中添加“指令计数”功能，统计接收的“LED\_ON”/“LED\_OFF”指令次数，通过串口显示“ON:5, OFF:3”，提交截图。

### (5) 反思

学生易在“组播地址与单播地址重复”上出错（如组播地址设为0x0001，与终端1地址冲突），下次课需强调“组播地址选0x1000以上，避开终端单播地址范围”；

部分学生对“NLME\_JoinMulticastGroup()函数返回值判断”忽略（未判断是否加入成功），下次课需添加返回值检查代码（如if(NLME\_JoinMulticastGroup(...) != SUCCESS) { /\* 加入失败处理 \*/ }）。

## 第十周：点播通讯 - 无线通讯（11月10日 - 11月14日）

### 1. 教学课题和内容

课题：WSN 点播通讯原理与点对点无线控制

**内容：**①点播通讯的定义（一对一通信，终端节点间直接通信，无需协调器转发）与核心条件（已知目标节点的网络地址）；②Zstack 点播配置（节点地址获取方法：手动配置、协调器分配后查询）；③点播数据收发代码实现（源节点发送点播指令，目标节点接收后执行动作）；④实验：终端节点间点播控制 LED（终端 1 发送“LED\_ON”/“LED\_OFF”点播指令到终端 2，终端 2 接收后控制自身 LED，实现“点对点”控制）。

## 2. 教学课时

总学时 4 学时（理论 2 学时 + 实验 2 学时）

## 3. 学生已有知识分析

**已有基础：**学生已掌握协调器与终端的单播 / 组播通信，熟悉 Zstack 协议栈的 AF\_DataRequest () 函数；了解“网络地址标识节点”，对“终端节点间通信需知道对方地址”有初步认知。

**知识缺口：**不熟悉终端节点间地址获取的具体方法（如通过协调器查询或手动预设）；对“点播通讯与单播通讯的区别”（点播是终端间直接通信，单播是终端→协调器）陌生；缺乏“终端节点接收点播指令后的中断处理”经验，易混淆“接收协调器指令”与“接收终端指令”。

## 4. 教学目标

**知识目标：**理解点播通讯的“一对一终端间通信”特性，掌握节点地址获取的 2 种方法，明确点播收发的代码逻辑。

**能力目标：**能获取终端节点的网络地址，能编写点播数据发送与接收代码，实现终端间 LED 控制。

**素养目标：**培养“点对点精准通信”的思维，通过地址验证与指令加密培养“数据安全”的意识。

## 5. 教学重点

终端节点地址获取（手动预设地址、通过协调器查询地址表）；

点播通讯代码编写（源节点设目标地址为终端地址，目标节点接收并执行指令）。

## 6. 教学难点

终端节点间直接通信的路由问题（若距离较远，需路由器转发，本次实验简化为近距离直接通信）；

点播指令的安全性（防止非目标节点接收，可通过指令加密或地址验证）。

## 7. 课程思政

结合“点播通讯的安全性”（指定目标地址，只有目标节点接收，避免数据泄露），讲解物联网数据安全的重要性（如智能家居中“手机控制卧室灯，仅卧室节点接收指令”），培养学生的安全意识与责任意识；

类比“点播通讯的‘精准传达’”与“生活中‘私人信息一对一沟通，不公开传播’”，引导学生树立“尊重隐私、保护信息”的道德观念。

## 8. 教学方法

理论教学：讲授法（结合点播通讯拓扑图）、案例分析法（分析终端间点播控制的应用案例）；

实验教学：示范操作法（教师演示地址获取与点播代码编写）、分组实操法（2组各做1个终端节点，验证点对点控制）。

## 9. 教学程序

### (1) 导入 (20 分钟)

场景导入：“智慧家居中，客厅的终端节点想控制卧室的 LED，不需要经过协调器转发，如何实现两个终端节点直接通信？需要知道什么关键信息？”引出点播通讯；

实物演示：展示“终端 1 按下按键，终端 2 的 LED 点亮”的效果，提问“这个过程中，终端 1 如何知道终端 2 的地址？终端 2 如何判断指令是发给自己的？”

### (2) 教授 (100 分钟)

理论讲解（40 分钟）：

点播通讯原理：①定义：两个终端节点（非协调器）之间的一对一直接通信，数据不经过协调器转发，适用于“终端间近距离、精准控制”场景；②核心条件：源节点必须知道目标节点的网络地址（如终端 2 地址为 0x0002）；③与单播的区别：单播是“终端→协调器”，点播是“终端→终端”；

节点地址获取方法：①手动预设：在代码中直接定义目标地址（如 #define TARGET\_ADDR 0x0002，适用于固定节点）；②协调器查询：协调器维护节点地址表，终端节点可向协调器发送查询请求，获取其他节点地址（适用于动态节点）；本次实验用手动预设（简化操作）；

点播收发流程：①源节点（终端 1）：发送指令时，目标地址设为目标终端地址（0x0002），簇 ID 设为 0x0003（点播专用簇 ID，区别于终端→协调器的簇 ID）；②目标节点（终端 2）：注册与源节点相同的端点号（0x01）与簇 ID（0x0003），接收数据时验证源地址（可选），解析指令并控制自身 LED。

实验操作（60 分钟）：

节点配置（10 分钟）：①终端 1（源节点）：预设目标地址为 0x0002（终端 2 地址），按键 1 控制发送“LED\_ON”，按键 2 控制发送“LED\_OFF”；②终端 2（目标节点）：预设自身地址为 0x0002（在协议栈工程中修改，如在 f8wConfig.cfg 中设 NWK\_ADDR=0x0002），LED 接 P1.0；③协调器：仅用于组建网络（终端 1、2 加入网络），不参与数据转发；

代码编写（35 分钟）：①终端 1（GenericApp.c）：在按键中断中，调用 AF\_DataRequest()，目标地址设为 TARGET\_ADDR（0x0002），簇 ID 设为 0x0003，数据为“LED\_ON”/“LED\_OFF”；②终端 2（GenericApp.c）：在 GenericApp\_Init() 中注册簇 ID 0x0003（GenericApp\_ClusterList[0]=0x0003;），接收数据时解析指令，控制 P1.0 LED（如收到“LED\_ON”则亮）；

组网与验证（15 分钟）：①协调器上电创建网络；②终端 2 上电加入网络（地址 0x0002，串口确认）；③终端 1 上电加入网络，按下按键 1，观察终端 2 LED 是否点亮；按下按键 2，观察是否熄灭；若失败，检查地址是否正确、簇 ID 是否匹配。

### (3) 总结 (15 分钟)

点播流程梳理：协调器组网→终端 2 加入（地址 0x0002）→终端 1 加入（预设目标地址 0x0002）→终端 1 发送点播指令→终端 2 接收执行；

核心要点：地址必须准确（源节点预设目标地址，目标节点地址需与预设一致），簇 ID 必须匹配（源节点发送簇 ID = 目标节点注册簇 ID），近距离通信（避免路由问题，简化实验）。

### (4) 作业

基础题：实现“终端 2 发送‘LED\_TOGGLE’点播指令到终端 1，终端 1 接收后翻转自身 LED”，完成双向点播控制；

拓展题：在终端 1 中添加“地址查询”功能，向协调器发送查询请求，获取终端 2 的地址（而非手动预设），提交代码与测试视频。

## (5) 反思

学生易在“目标节点地址设置”上出错（如终端 2 实际地址为 0x0003，终端 1 预设 0x0002），下次课可通过“协调器串口打印节点地址表”（如 Z-Tool 读取），帮助学生获取准确地址；部分学生对“簇 ID 匹配”理解不足（源节点用 0x0001，目标节点注册 0x0003，导致无法接收），下次课需强调“发送簇 ID = 接收簇 ID”的必要性。

# 第十一周：温度传感器 DS18B20 的使用（11 月 17 日 - 11 月 21 日）

## 1. 教学课题和内容

课题：DS18B20 温度传感器原理与 CC2530 数据采集

内容：①DS18B20 的工作模式（单总线通信）与核心特性（高精度，测量范围 -55°C~125°C，分辨率 0.0625°C）；②DS18B20 数据采集时序（复位信号、存在信号、ROM 指令、功能指令、数据读取）；③温度数据格式与转换（16 位数据，整数部分与小数部分分离计算）；④实验：DS18B20 温度采集与无线传输（终端节点采集温度，通过点播发送到另一终端节点，目标节点接收后通过串口显示“Temp: 25.5°C”）。

## 2. 教学课时

总学时 2 学时（理论 1 学时 + 实验 1 学时）

## 3. 学生已有知识分析

已有基础：学生已掌握 DHT11 温湿度采集（单总线），熟悉 CC2530 GPIO 操作与单总线时序控制；了解“传感器采集数据需遵循严格时序”，对“温度数据含整数与小数部分”有初步认知。

知识缺口：不熟悉 DS18B20 的 ROM 指令（如跳过 ROM 指令 0xCC，无需读取设备 ID）与功能指令（如温度转换指令 0x44）；对“16 位温度数据的解析”（高 8 位与低 8 位组合，符号位判断正负温度）陌生；缺乏“DS18B20 与 DHT11 的时序差异”（DS18B20 时序更严格，需精准延时）处理经验。

## 4. 教学目标

知识目标：理解 DS18B20 的高精度特性与单总线时序，掌握温度数据解析方法，明确 DS18B20 与 DHT11 的差异。

能力目标：能独立连接 DS18B20 与 CC2530，能编写数据采集与解析代码，实现温度采集与无线点播传输。

素养目标：培养“高精度数据采集”的工程意识，通过数据解析培养“严谨处理细节”的科学态度。

## 5. 教学重点

DS18B20 数据采集时序（复位→存在→ROM 指令→功能指令→数据读取）；  
16 位温度数据解析（整数部分、小数部分计算，正负温度判断）。

## 6. 教学难点

DS18B20 严格的时序控制（复位信号低电平需 480~960 $\mu$ s，存在信号检测窗口窄）；  
负温度数据解析（符号位为 1 时，需对数据取反加 1 后计算）。

## 7. 课程思政

对比 DS18B20（高精度，适用于工业温控）与 DHT11（低成本，适用于普通环境），讲解“根据应用场景选择合适传感器”的工程思维，引导学生树立“实用、适配”的设计理念；  
展示 DS18B20 在“工业设备温度监测”中的应用（如电机轴承温度监测，超 50 $^{\circ}$ C 报警），说明高精度数据对“设备故障预警”的重要性，培养学生的责任意识。

## 8. 教学方法

理论教学：讲授法（结合 DS18B20 时序波形图与数据格式表）、对比分析法（DS18B20 与 DHT11 的时序、精度、成本差异）；

实验教学：示范操作法（教师演示时序代码与数据解析代码）、分组实操法（2 人一组，一人负责采集，一人负责无线传输）。

## 9. 教学程序

### (1) 导入 (15 分钟)

问题导入：“DHT11 温度测量精度是  $\pm 2^{\circ}$ C，无法满足工业设备（如电机） $\pm 0.5^{\circ}$ C 的精度要求，用什么传感器能实现高精度温度采集？”引出 DS18B20；

实物对比：展示 DS18B20 与 DHT11，说明“DS18B20 体积更小、精度更高，但时序更严格”，提问“它的单总线时序与 DHT11 有什么不同？”

### (2) 教授 (45 分钟)

理论讲解（20 分钟）：

DS18B20 工作原理：单总线通信，供电 3.3-5V，测量范围 -55 $^{\circ}$ C~125 $^{\circ}$ C，分辨率 0.0625 $^{\circ}$ C，需通过严格时序传输数据；

采集时序：①复位信号：CC2530 拉低 DATA 线 480~960 $\mu$ s，再拉高 15~60 $\mu$ s；②存在信号：DS18B20 拉低 DATA 线 60~240 $\mu$ s，再拉高，CC2530 需在拉高前检测到低电平；③ROM 指令：发送跳过 ROM 指令 0xCC（无需读取 DS18B20 的唯一 ID，简化操作）；④功能指令：发送温度转换指令 0x44（DS18B20 开始测量，需等待 100ms 以上）；⑤数据读取：发送读数据指令 0xBE，连续读取 8 字节数据，其中第 1、2 字节为温度数据（低 8 位、高 8 位）；

温度数据解析：①16 位数据格式：bit15 为符号位（0 $\rightarrow$ 正，1 $\rightarrow$ 负），bit14-bit4 为整数部分，bit3-bit0 为小数部分；②计算方法：正温度 =（高 8 位  $\ll$  8 | 低 8 位） $\times$  0.0625；负温度 = -[（高 8 位  $\ll$  8 | 低 8 位）取反 + 1] $\times$  0.0625（如数据 0xFFF8 $\rightarrow$ 取反 0x0007 $\rightarrow$ 加 1 0x0008 $\rightarrow$ 8 $\times$  0.0625=0.5 $^{\circ}$ C $\rightarrow$ 负温度 -0.5 $^{\circ}$ C）。

实验操作（25 分钟）：

电路连接（8 分钟）：①DS18B20 VCC 接 CC2530 3.3V；②GND 接 GND；③DATA 接 CC2530 P0.5；④DATA 线串联 4.7k $\Omega$  上拉电阻；

代码编写（12 分钟）：①DS18B20 复位与存在检测函数（uint8 DS18B20\_Reset(void){ /\* 实现复位与存在信号检测 \*/ }）；②温度采集函数（float DS18B20\_ReadTemp(void){ / 发送指令、读取数据、解析温度 \*/ }）；③无线点播传输：终端 1（采集节点）将温度格式化为“25.5”，发送到终端 2（目标节点，地址 0x0002）；终端 2 接收后通过串口显示“Temp: 25.5 $^{\circ}$ C”；

功能验证（5分钟）：烧录代码，上电后观察终端2串口，若显示“Temp: 25.5°C”则成功；若数据异常（如 -127°C），检查时序与上拉电阻。

### (3) 总结 (10 分钟)

时序与解析梳理：复位→存在→指令→数据读取→解析（整数+小数+正负）；

DS18B20 与 DHT11 对比：①精度：DS18B20 (0.0625°C) > DHT11 (±2°C)；②时序：DS18B20 更严格；③成本：DS18B20 更高；④应用：DS18B20 适用于高精度场景，DHT11 适用于普通场景。

### (4) 作业

基础题：修改代码，当温度 > 30°C 时，终端 2 LED 点亮并串口发送“High Temp: 30.5°C”；

拓展题：实现“终端 1 采集温度，终端 2 采集湿度（DHT11），两者通过点播互传数据，各自串口显示“Temp:25.5°C, Hum:60%”，提交测试截图。

### (5) 反思

学生易在“负温度解析”上出错（忘记取反加1），下次课可提供“正负温度解析代码模板”，标注关键步骤；

部分学生对“DS18B20 温度转换等待时间”把控不足（未等 100ms 就读取数据，导致数据错误），下次课需在代码中添加延时函数注释（如 // 等待温度转换完成，至少 100ms Delay\_ms (150);）。

## 第十二周：温湿度传感器 DHT11（进阶）（11月24日 - 11月28日）

### 1. 教学课题和内容（续）

③异常报警逻辑（续）：设置阈值，如湿度 < 50% 或 > 80%、温度 < 10°C 或 > 35°C 时，发送报警指令；协调器接收后区分节点类型（大棚节点 / 仓库节点），执行对应报警动作（点亮不同颜色 LED、串口输出报警信息）；

④实验（续）：DHT11 进阶应用（终端节点周期性采集温湿度，无线发送到协调器；协调器接收后，若数据异常则点亮报警 LED，并通过串口发送“Alarm: Humidity Low!”）。

### 2. 教学程序（续）

#### (2) 教授（续，100 分钟）

理论讲解（续，40 分钟）：

异常报警逻辑（续）：①阈值设置：按场景定义（大棚节点：H=50%-80%，T=15°C-35°C；仓库节点：H=40%-70%，T=10°C-30°C），在代码中通过宏定义区分（如 #define NODE\_TYPE 1 //1 = 大棚，2 = 仓库）；②报警触发：终端节点采集数据后，先判断是否超出自身阈值，若异常则在发送数据时添加“ALM”标识（如“1,45,25,ALM”表示大棚节点湿度 45% 异常）；③协调器处理：接收数据时，先解析节点类型与“ALM”标识，若异常则点亮对应报警 LED（大棚节点异常→红色 LED 亮，仓库节点异常→黄色 LED 亮），并串口输出具体报警信息（如“Alarm: Node 1 (Greenhouse) Humidity Low! H=45%”）。

实验操作（60 分钟）：

节点配置（15分钟）：①终端节点1（大棚节点）：NODE\_TYPE=1，DHT11接P0.4，定时器3设为5秒周期；②终端节点2（仓库节点）：NODE\_TYPE=2，DHT11接P0.5，定时器3设为6秒周期；③协调器：红色LED（P1.1）对应大棚报警，黄色LED（P1.2）对应仓库报警，串口波特率9600bps；

代码编写（30分钟）：

①终端节点：在定时器中断中调用DHT11采集函数，采集后通过if(NODE\_TYPE==1 && (hum<50||hum>80||temp<15||temp>35))判断异常，若异常则格式化数据为“1,45,25,ALM”，调用AF\_DataRequest()发送到协调器（地址0x0000）；

②协调器：在GenericApp\_ProcessEvent()中接收数据，通过strstr((char\*)pkt->cmd.Data, "ALM")判断是否异常，解析节点类型（如第一个数字为1则是大棚节点），若大棚湿度低则P1\_1=0（红色LED亮），串口发送“Alarm: Node 1 (Greenhouse) Humidity Low! H=45%”；

功能验证（15分钟）：①各节点上电，协调器确认节点加入（串口显示“Node 0x0001 (Greenhouse) Joined”）；②用加湿器模拟大棚湿度升高至85%，观察协调器红色LED是否点亮，串口是否显示湿度高报警；③用风扇降温模拟仓库温度降至8°C，观察黄色LED是否点亮，验证报警逻辑。

### (3) 总结（15分钟）

进阶逻辑梳理：周期性采集（定时器）→数据压缩（简洁格式）→异常判断（阈值）→报警联动（LED+串口）；

核心要点：节点类型区分（宏定义）、异常标识（ALM）、协调器联动响应（不同LED），确保“采集-判断-报警”闭环。

### (4) 作业

基础题：修改仓库节点阈值（H=35%-75%，T=12°C-28°C），重新编写代码，验证报警功能；

拓展题：在终端节点中添加“报警解除”逻辑（数据恢复正常后，发送“ALM\_CLR”指令），协调器接收后熄灭报警LED，提交测试视频。

### (5) 反思

学生易混淆“节点类型宏定义”（如将NODE\_TYPE设为1却按仓库阈值判断），下次课可提供“节点配置模板”（标注大棚/仓库对应的阈值与宏定义值）；

部分学生对“定时器中断与DHT11采集的时序冲突”（中断打断采集）处理不当，需在代码中添加“采集标志位”（如if(!isCollecting){isCollecting=1;采集;isCollecting=0;}）避免冲突。

## 第十三周：按键无线控制台灯和LED灯（12月1日-12月5日）

### 1. 教学课题和内容

课题：WSN多设备联动控制（按键→协调器→台灯/LED）

内容：①多设备联动逻辑（终端按键节点→协调器→执行节点（台灯/LED），协调器作为“中控”转发指令）；②指令格式设计（区分控制对象与动作，如“DEV\_LED,ON”表示控制LED点亮，“DEV\_LAMP,OFF”表示控制台灯熄灭）；③执行节点驱动（LED直接控制，台灯通过继电器驱动（CC2530 GPIO控制继电器吸合/断开））；④实验：按键无线控制多设备（终端1按键发送

“DEV\_LED,ON”→协调器转发→LED 节点点亮 LED；终端 2 按键发送 “DEV\_LAMP,OFF”→协调器转发→台灯节点断开台灯电源）。

## 2. 教学课时

总学时 2 学时（理论 1 学时 + 实验 1 学时）

## 3. 学生已有知识分析

**已有基础：**学生已掌握 Zstack 协议栈下的无线收发（终端→协调器），熟悉 CC2530 GPIO 控制（LED、按键）；了解“指令格式需包含控制对象与动作”，对“继电器用于控制大功率设备”有初步认知。

**知识缺口：**不熟悉“协调器转发指令”的逻辑（需解析指令中的控制对象，转发到对应执行节点）；对“继电器驱动电路”（需三极管放大电流，避免 GPIO 直接驱动）陌生；缺乏“多执行节点地址管理”经验（协调器需存储各执行节点地址）。

## 4. 教学目标

**知识目标：**理解多设备联动的“按键 - 协调器 - 执行”三层逻辑，掌握指令格式设计与继电器驱动原理，明确协调器转发机制。

**能力目标：**能设计控制指令格式，能搭建继电器驱动电路，能编写协调器转发代码，实现按键控制台灯与 LED。

**素养目标：**培养“系统联动设计”的全局思维，通过继电器安全驱动培养“大功率设备控制的安全意识”。

## 5. 教学重点

多设备联动逻辑（协调器解析指令→转发到对应执行节点）；  
继电器驱动电路搭建（CC2530 GPIO→三极管→继电器→台灯）。

## 6. 教学难点

协调器指令转发的地址管理（通过“控制对象 - 地址”映射表，如 DEV\_LED 对应地址 0x0003，DEV\_LAMP 对应 0x0004）；  
继电器吸合 / 断开的电平控制（避免电平反转导致动作错误）。

## 7. 课程思政

展示“WSN 多设备联动在智能家居中的应用”（如“开门→灯光自动亮 + 窗帘自动开”），说明技术对“便捷生活、提升幸福感”的作用，激发学生对物联网行业的创新热情；

强调“继电器驱动的安全性”（大功率设备需通过继电器隔离，避免 GPIO 烧毁），类比“生活中‘高压危险需隔离’”，引导学生树立“安全第一”的工程观念。

## 8. 教学方法

理论教学：讲授法（结合多设备联动拓扑图）、案例分析法（分析智能家居联动场景）；

实验教学：示范操作法（教师演示继电器电路搭建与转发代码编写）、分组实操法（4 组分工：1 组按键节点、1 组 LED 节点、1 组台灯节点、1 组协调器）。

## 9. 教学程序

### (1) 导入 (15 分钟)

场景导入：“回家后想同时打开客厅 LED 和卧室台灯，需要分别按两个按键，能不能通过一个按键节点发送指令，让协调器自动控制多个设备？”引出多设备联动；

实物演示：展示“终端按键→协调器→LED 亮 + 台灯亮”的联动效果，提问“协调器如何知道指令该发给 LED 还是台灯？需要什么特殊指令格式？”

### (2) 教授 (45 分钟)

理论讲解 (20 分钟)：

多设备联动逻辑：①三层架构：按键节点（指令发送，如终端 1、2）→协调器（指令解析与转发，地址 0x0000）→执行节点（动作执行，LED 节点 0x0003、台灯节点 0x0004）；②指令格式：“控制对象，动作”（如“DEV\_LED,ON”“DEV\_LAMP,OFF”），控制对象用固定标识（DEV\_LED/DEV\_LAMP），动作用 ON/OFF；

协调器转发机制：①地址映射表：在协调器代码中定义 `uint16 devAddrMap[2] = {0x0003, 0x0004};`（DEV\_LED 对应 0x0003，DEV\_LAMP 对应 0x0004）；②解析与转发：接收按键节点指令后，用 `strstr((char*)data, "DEV_LED")` 判断控制对象，从映射表中获取执行节点地址，调用 `AF_DataRequest()` 转发指令；

继电器驱动原理：①电路组成：CC2530 P1.3→NPN 三极管（S8050）基极（串联 1kΩ 限流电阻）→三极管集电极→继电器线圈→VCC 5V，继电器常开触点→台灯电源；②控制逻辑：P1.3=1→三极管导通→继电器吸合→台灯通电；P1.3=0→三极管截止→继电器断开→台灯断电。

实验操作 (25 分钟)：

电路搭建 (10 分钟)：①LED 节点：LED 接 P1.0，地址 0x0003；②台灯节点：搭建继电器驱动电路（P1.3→三极管→继电器→台灯），地址 0x0004；③按键节点 1：按键接 P0.0，发送“DEV\_LED,ON”/“DEV\_LED,OFF”；④按键节点 2：按键接 P0.1，发送“DEV\_LAMP,ON”/“DEV\_LAMP,OFF”；

代码编写 (10 分钟)：①协调器：编写指令解析函数，如 `if(strstr(data, "DEV_LED")) { dstAddr=0x0003; } else if(strstr(data, "DEV_LAMP")) { dstAddr=0x0004; }`，调用 `AF_DataRequest()` 转发；②执行节点：LED 节点接收“DEV\_LED,ON”则 P1\_0=0，台灯节点接收“DEV\_LAMP,ON”则 P1\_3=1；

功能验证 (5 分钟)：①各节点上电，协调器存储执行节点地址；②按下按键节点 1 的“ON”键，观察 LED 节点是否点亮；③按下按键节点 2 的“OFF”键，观察台灯是否断电，验证联动逻辑。

### (3) 总结 (10 分钟)

联动流程梳理：按键节点发送指令→协调器解析对象→查询地址映射→转发指令→执行节点动作；

核心要点：指令格式统一（对象 + 动作）、地址映射准确（避免转发错误）、继电器驱动安全（限流电阻不可少）。

### (4) 作业

基础题：添加“DEV\_FAN”（风扇）执行节点（地址 0x0005），修改协调器映射表与按键节点指令，实现按键控制风扇启停；

拓展题：在协调器中添加“定时控制”功能，每天 18:00 自动发送“DEV\_LED,ON”指令，提交代码与测试截图。

## (5) 反思

学生易在“地址映射表与控制对象匹配”上出错（如 DEV\_LED 对应 0x0004），下次课可在协调器代码中用“枚举”定义控制对象（enum {DEV\_LED=0, DEV\_LAMP=1};），避免混淆；部分学生继电器驱动电路中忘记加限流电阻，导致三极管烧毁，需提前准备“继电器驱动模块”（已焊接好电阻与三极管），减少硬件故障。

# 第十四周：串口透传之节点相互收发信息（12月8日 - 12月12日）

## 1. 教学课题和内容

**课题：**WSN 节点串口透传原理与双向信息交互

**内容：**①串口透传定义（节点串口接收的电脑数据，通过无线透明传输到另一节点，再通过串口输出到电脑，实现“电脑1→节点1→节点2→电脑2”的双向通信）；②Zstack 协议栈串口透传配置（串口与无线的波特率匹配、数据帧封装（添加透传标识“TT”，避免与控制指令混淆））；③透传数据完整性保障（添加校验位，如异或校验，避免数据丢失）；④实验：节点双向串口透传（电脑1通过串口向节点1发送“Hello from PC1”，节点1无线透传到节点2，节点2通过串口输出到电脑2；反之实现电脑2→节点2→节点1→电脑1的透传）。

## 2. 教学课时

总学时 4 学时（理论 2 学时 + 实验 2 学时）

## 3. 学生已有知识分析

**已有基础：**学生已掌握 CC2530 串口 UART0 发送 / 接收（如接收电脑数据）、Zstack 无线收发；了解“透传需保持数据格式不变”，对“波特率匹配”（串口与无线波特率一致）有初步认知。

**知识缺口：**不熟悉“串口接收中断与无线发送的协同”（串口接收数据后需立即触发无线发送，避免数据缓存溢出）；对“透传数据校验”（如异或校验）陌生；缺乏“双向透传的地址管理”经验（节点需知道对方透传节点地址）。

## 4. 教学目标

**知识目标：**理解串口透传的“透明传输”特性，掌握透传配置与数据校验方法，明确双向透传的地址与波特率匹配要求。

**能力目标：**能配置串口与无线波特率，能编写串口接收中断与无线透传代码，实现节点双向串口透传与数据校验。

**素养目标：**培养“数据透明传输”的工程思维，通过校验机制培养“保障数据完整性”的严谨态度。

## 5. 教学重点

串口透传配置（串口与无线波特率均设为 9600bps，数据帧添加透传标识）；

串口接收中断编写（接收电脑数据后，触发无线透传）。

## 6. 教学难点

串口接收数据的缓存处理（用数组缓存多字节数据，通过结束符“\n”判断数据接收完成）；透传数据校验（异或校验位计算与验证，如数据“Hello”的校验位 =  $H^e^l^l^o$ ）。

## 7. 课程思政

展示“串口透传在工业监控中的应用”（如远程 PLC 通过 WSN 节点透传，读取车间传感器数据），说明技术对“工业自动化、无人值守”的支撑作用，培养学生的产业使命感；强调“透传数据完整性”（校验位缺失可能导致设备误操作），类比“生活中‘重要文件需核对完整性’”，引导学生树立“重视细节、保障可靠”的意识。

## 8. 教学方法

理论教学：讲授法（结合串口透传流程图）、公式推导法（演示异或校验位计算）；实验教学：示范操作法（教师演示串口接收中断与透传代码）、分组实操法（2 组各做 1 个透传节点，连接 2 台电脑验证双向透传）。

## 9. 教学程序

### (1) 导入 (20 分钟)

问题导入：“电脑 1 想向电脑 2 发送数据，但两台电脑距离 100 米，无法直接连线，如何通过两个 WSN 节点实现‘无线搭桥’？”引出串口透传；

对比导入：展示“直接连线通信 vs 串口透传通信”的差异，提问“透传为什么能‘透明’传输数据？需要解决哪些问题（如波特率、校验）？”

### (2) 教授 (100 分钟)

理论讲解（40 分钟）：

串口透传原理：①定义：节点 1 串口接收电脑 1 数据后，不修改数据内容，直接通过无线发送到节点 2，节点 2 串口将数据输出到电脑 2，实现“电脑 1 - 电脑 2”的无线通信，数据格式完全透明；②核心要求：串口与无线波特率一致（均设为 9600bps）、数据帧添加透传标识（如“TT:Hello”，避免与控制指令混淆）；

串口接收中断配置：①UART0 接收中断使能（ $IEN0 |= 0x04$ ;  $U0CSR |= 0x40$ ；），接收数据时触发中断；②数据缓存：定义数组 `uint8 uartBuf[32]`; `uint8 uartLen=0`；，中断服务函数中 `uartBuf[uartLen++]=U0DBUF`；，当接收到结束符“\n”时，触发无线透传；

数据校验方法：①异或校验：计算数据（除校验位外）的异或值，添加到数据末尾（如“TT:Hello”的校验位 =  $H^e^l^l^o=0x48^0x65^0x6C^0x6C^0x6F=0x0A$ ），数据格式为“TT:Hello,0x0A”；②校验验证：接收方计算数据部分的异或值，与校验位对比，一致则数据有效。

实验操作（60 分钟）：

节点配置（15 分钟）：①节点 1（透传发送端）：UART0 波特率 9600bps，无线地址设为 0x0005，串口接电脑 1；②节点 2（透传接收端）：UART0 波特率 9600bps，无线地址设为 0x0006，串口接电脑 2；两节点互设为透传目标地址；

代码编写（30 分钟）：①节点 1：编写 UART0 接收中断服务函数，接收“Hello from PC1\n”后，添加透传标识与校验位（“TT:Hello from PC1,0xXX”），调用 `AF_DataRequest()` 发送到节点 2；②节点 2：接收数据后，解析透传标识“TT”，验证校验位，若有效则通过 UART0 发送到电脑 2（调用 `UART0_SendStr()`）；

功能验证（15 分钟）：①电脑 1 串口助手发送“Hello from PC1”，观察节点 1 是否透传，电脑 2 串口是否接收并显示；②电脑 2 发送“Hi from PC2”，验证反向透传；③故意修改校验位，观察节点 2 是否提示“Data Error”，验证校验逻辑。

### (3) 总结 (15 分钟)

透传流程梳理：电脑 1 串口发送→节点 1 接收中断→添加标识与校验→无线透传→节点 2 接收校验→串口输出→电脑 2；

核心要点：波特率一致（9600bps）、标识区分（TT）、校验保障（异或），确保“透明 + 可靠”传输。

### (4) 作业

基础题：修改代码，实现透传数据长度扩展到 64 字节，验证长数据透传完整性；

拓展题：在透传中添加“数据重传”功能（接收方校验失败时，发送“RETRY”指令，发送方重新透传），提交测试视频。

### (5) 反思

学生易在“串口接收中断的结束符判断”上出错（如用“\r”而非“\n”），下次课需统一串口助手的“发送结束符”设置（勾选“发送换行”）；

部分学生对“异或校验计算”理解困难，可提供“校验计算工具”（Excel 表格自动计算），帮助验证校验位。

## 第十五周：基于 GenericApp 无线收发实验（12 月 15 日 - 12 月 19 日）

### 1. 教学课题和内容

课题：Zstack GenericApp 工程二次开发与无线收发优化

内容：①GenericApp 工程结构（应用层文件 GenericApp.c/h、协议栈依赖文件）；②GenericApp 核心函数（GenericApp\_Init () 初始化、GenericApp\_ProcessEvent () 事件处理、GenericApp\_MessageMSGCB () 消息回调）；③基于 GenericApp 的功能扩展（添加 DHT11 温湿度采集，将数据融入无线收发帧）；④实验：GenericApp 二次开发（修改 GenericApp 工程，实现“终端节点采集温湿度→无线发送到协调器→协调器串口显示 + LED 指示（温度 > 30°C 亮红灯）”）。

### 2. 教学课时

总学时 2 学时（理论 1 学时 + 实验 1 学时）

### 3. 学生已有知识分析

已有基础：学生已熟悉 Zstack 协议栈基本操作（工程创建、节点配置），掌握 DHT11 采集与无线收发；了解“GenericApp 是协议栈自带的基础应用工程”，对“事件处理函数”（如 SYS\_EVENT\_MSG）有初步认知。

知识缺口：不熟悉 GenericApp 工程的文件依赖关系（如修改 GenericApp.c 需关联的头文件）；对“GenericApp\_ProcessEvent () 中的事件优先级”（系统事件优先于应用事件）陌生；缺乏“在 GenericApp 中整合多模块功能”（采集 + 收发 + 显示）的经验。

### 4. 教学目标

**知识目标：**理解 GenericApp 工程结构与核心函数，掌握基于 GenericApp 的功能扩展方法，明确事件处理的优先级逻辑。

**能力目标：**能独立修改 GenericApp 工程文件，能在事件处理函数中添加采集与收发逻辑，实现多模块功能整合。

**素养目标：**培养“基于现有工程二次开发”的创新思维，通过代码模块化（采集、收发、显示分离）培养“规范编程”的习惯。

## 5. 教学重点

GenericApp 核心函数（GenericApp\_Init () 初始化外设、GenericApp\_ProcessEvent () 处理无线接收事件）；

多模块整合（DHT11 采集→无线发送→协调器显示 + LED 控制）。

## 6. 教学难点

GenericApp 事件处理的优先级（如 SYS\_EVENT\_MSG 事件需优先处理，避免消息丢失）；

工程文件依赖问题（如添加 DHT11 采集需包含“dht11.h”头文件，需在工程选项中添加路径）。

## 7. 课程思政

讲解“二次开发”的行业价值（基于成熟工程快速迭代功能，降低开发成本），类比“站在巨人肩膀上创新”，引导学生树立“高效利用现有资源、持续优化”的开发理念；

展示“GenericApp 在物联网项目中的应用”（如快速搭建 WSN 温湿度监测原型），说明“基础工程是项目落地的重要支撑”，培养学生“重视基础、夯实技能”的意识。

## 8. 教学方法

理论教学：讲授法（结合 GenericApp 工程文件结构图）、代码分析法（逐行讲解核心函数）；

实验教学：示范操作法（教师演示工程修改与函数添加）、分组实操法（2 人一组，共同修改 GenericApp 工程）。

## 9. 教学程序

### (1) 导入 (15 分钟)

问题导入：“之前我们创建的工程都是从零开始，协议栈自带的 GenericApp 工程已经实现了基础无线收发，如何在它的基础上快速添加温湿度采集功能？”引出 GenericApp 二次开发；

演示导入：打开 GenericApp 工程，展示“终端节点发送‘Hello’→协调器接收显示”的基础功能，提问“如何修改代码，让发送的数据从‘Hello’变成‘H=60%,T=25°C’？”

### (2) 教授 (45 分钟)

理论讲解（20 分钟）：

GenericApp 工程结构：①核心文件：GenericApp.c（应用逻辑）、GenericApp.h（宏定义与函数声明）；②依赖文件：协议栈内核文件（如 OSAL.h、AF.h）、硬件驱动文件（如 ioCC2530.h）；③工程配置：在 IAR 中需设置“Include Paths”，添加自定义头文件（如 dht11.h）路径；

核心函数解析：①GenericApp\_Init ()：初始化端点号（GenericApp\_EndPointDesc.endPoint=0x01）、注册簇 ID（GenericApp\_ClusterList [0]=0x0001）、

初始化外设（DHT11、LED、串口）；②GenericApp\_ProcessEvent()：处理事件，当事件为 SYS\_EVENT\_MSG 时，调用 GenericApp\_MessageMSGCB() 处理无线接收消息；③GenericApp\_MessageMSGCB()：消息回调函数，解析接收数据，执行对应动作（如协调器串口显示、LED 控制）；

功能扩展逻辑：在 GenericApp\_Init() 中初始化 DHT11，在 GenericApp\_ProcessEvent() 中添加“周期性采集事件”（如每 5 秒触发一次），采集后调用 AF\_DataRequest() 发送数据，协调器在 GenericApp\_MessageMSGCB() 中解析温湿度，控制 LED。

实验操作（25 分钟）：

工程准备（5 分钟）：①打开 Zstack 协议栈的 GenericApp 工程（路径：ZStack-CC2530-2.5.1a\Projects\zstack\Samples\GenericApp\CC2530DB）；②配置协调器（DEVICE\_TYPE=COORDINATOR）与终端节点（DEVICE\_TYPE=END\_DEVICE）工程；

代码修改（15 分钟）：①终端节点 GenericApp.c：在 GenericApp\_Init() 中添加 DHT11\_Init()；在 GenericApp\_ProcessEvent() 中添加“if(events & GENERICAPP\_SEND\_EVT) { 采集温湿度；格式化数据；发送；return events ^ GENERICAPP\_SEND\_EVT; }”；②协调器 GenericApp.c：在 GenericApp\_MessageMSGCB() 中解析数据（如“T=32°C”），若温度 > 30°C 则 P1\_1=0（红灯亮），串口发送数据；

功能验证（5 分钟）：①烧录工程，终端节点上电采集并发送数据；②协调器接收后，观察串口是否显示“Humidity: 62%, Temperature: 31°C”，红灯是否点亮，验证二次开发效果。

### (3) 总结 (10 分钟)

二次开发流程：熟悉 GenericApp 结构→修改核心函数→添加模块功能→调试验证；

核心要点：工程路径配置（头文件路径）、事件优先级（系统事件优先）、代码模块化（采集、收发、显示函数分离）。

### (4) 作业

基础题：在 GenericApp 工程中添加按键控制，按下按键立即发送温湿度数据（而非周期性发送）；

拓展题：基于 GenericApp 实现“协调器向终端节点发送‘采集指令’→终端节点接收后采集并回复数据”的双向通信，提交工程代码。

### (5) 反思

学生易在“工程头文件路径配置”上出错（如添加 dht11.h 后编译报错“无法打开文件”），下次课可演示“在 IAR 中添加 Include Paths”的步骤（Project→Options→C/C++ Compiler→Preprocessor→Additional include directories）；

部分学生对“事件触发机制”理解困难，可通过“事件队列”动画（系统事件先入队先处理）强化认知。

## 第十六周：新建自己的 Zstack 工程（12 月 22 日 - 12 月 26 日）

### 1. 教学课题和内容

课题：Zstack 自定义工程创建与完整功能实现

内容：①Zstack 工程创建步骤（复制模板工程、修改工程名、配置芯片型号与协议栈版本）；②工程文件裁剪（删除 GenericApp 冗余代码，保留核心依赖）；③自定义工程功能设计（确定“终端

节点：DS18B20 采集 + 无线发送；协调器：无线接收 + 串口显示 + 继电器控制（温度 > 30°C 开风扇）”）；④实验：自定义 Zstack 工程（从模板创建工程，实现上述功能，完成编译、烧录与功能验证）。

## 2. 教学课时

总学时 4 学时（理论 2 学时 + 实验 2 学时）

## 3. 学生已有知识分析

**已有基础：**学生已掌握 GenericApp 二次开发，熟悉 Zstack 协议栈核心函数（初始化、事件处理）；了解“协议栈模板工程的位置”（如 Samples 文件夹下的 Template），对“工程配置选项”（如 DEVICE\_TYPE）有初步认知。

**知识缺口：**不熟悉自定义工程的文件裁剪逻辑（如哪些文件可删除，哪些需保留）；对“工程编译选项配置”（如堆栈大小、优化等级）陌生；缺乏“从 0 到 1 创建工程并整合完整功能”的经验。

## 4. 教学目标

**知识目标：**理解 Zstack 自定义工程的创建流程与文件裁剪方法，掌握工程编译配置要点，明确完整功能（采集 + 收发 + 控制）的实现逻辑。

**能力目标：**能独立从模板创建 Zstack 工程，能裁剪冗余文件并配置编译选项，能实现多模块完整功能整合。

**素养目标：**培养“从 0 到 1 自主开发”的工程能力，通过工程优化（裁剪冗余、配置堆栈）培养“高效、精简”的设计思维。

## 5. 教学重点

Zstack 自定义工程创建步骤（复制模板 → 修改配置 → 裁剪文件）；  
完整功能整合（DS18B20 采集 → 无线发送 → 协调器控制继电器）。

## 6. 教学难点

工程编译选项配置（堆栈大小设置：如 APP\_MAX\_MSGS=5，APP\_TX\_MAX\_EVT=3，避免内存溢出）；

文件裁剪后的依赖问题（如删除冗余.c 文件后需删除对应的头文件引用，避免编译报错）。

## 7. 课程思政

讲解“自主创建工程”对物联网项目开发的意义（根据需求定制功能，避免冗余代码占用资源），类比“自主设计方案 vs 套用模板”，引导学生树立“创新设计、按需定制”的工程观念；

展示“自定义工程在工业 WSN 项目中的应用”（如定制化的设备状态监测工程），说明“自主开发能力是行业核心竞争力”，激发学生的学习动力与职业自信。

## 8. 教学方法

理论教学：讲授法（结合工程创建流程图）、演示法（教师投屏展示每一步操作）；

实验教学：分步指导法（将工程创建拆分为“复制 → 配置 → 裁剪 → 编码 → 编译”5 步）、分组实操法（3 人一组，分工完成工程创建、代码编写、调试）。

## 9. 教学程序

### (1) 导入 (20 分钟)

问题导入：“之前基于 GenericApp 开发会有冗余代码（如不需要的按键逻辑），如何创建一个只包含‘温湿度采集 + 风扇控制’的精简工程？”引出自定义工程；

对比导入：展示“GenericApp 工程文件（15 个文件）vs 自定义工程文件（8 个文件）”，提问“自定义工程需要保留哪些核心文件？如何避免编译报错？”

### (2) 教授 (100 分钟)

理论讲解（40 分钟）：

自定义工程创建步骤：①复制模板：从 ZStack-CC2530-2.5.1a\Projects\zstack\Templates\SampleApp 复制工程到自定义目录，重命名为“WSN\_TempControl”；②修改工程配置：在 IAR 中打开工程，修改芯片型号为“CC2530F256”，在 f8wConfig.cfg 中设置 DEVICE\_TYPE（终端 = END\_DEVICE，协调器 = COORDINATOR）、设置堆栈大小（APP\_MAX\_MSGS=5，APP\_TX\_MAX\_EVT=3）；③文件裁剪：删除冗余文件（如 SampleApp.c/h 替换为自定义的“TempControl.c/h”，删除不需要的“hal\_led.c”外的其他驱动文件），保留核心文件（OSAL.c、AF.c、hal\_uart.c、TempControl.c）；

编译选项配置：①优化等级：设为“Low”（避免代码优化导致调试困难）；②输出文件：勾选“Generate HEX file”，设置输出路径；③错误检查：开启“Warning as error”，及时发现代码隐患；

完整功能逻辑：①终端节点（TempControl\_Init()）：初始化 DS18B20、UART0、无线端点，每 5 秒采集温度，发送数据“T=28.5°C”；②协调器（TempControl\_ProcessEvent()）：接收温度数据，若 > 30°C 则控制继电器（P1.3=1）开风扇，串口显示“Fan ON: T=31.2°C”。

实验操作（60 分钟）：

工程创建（20 分钟）：①按步骤复制模板、修改工程名、配置芯片与节点类型；②裁剪文件：删除 SampleApp.c/h，新建 TempControl.c/h，在工程中添加新文件；③配置堆栈与优化等级，编译工程，解决“头文件缺失”等基础报错；

代码编写（25 分钟）：①TempControl.c：编写初始化函数（DS18B20\_Init()、AF 端点注册）、事件处理函数（采集与发送）、消息回调函数（协调器接收与继电器控制）；②添加 DS18B20 驱动文件（ds18b20.c/h），配置头文件路径；

功能验证（15 分钟）：①编译生成 HEX 文件，烧录到终端与协调器；②终端节点采集温度（用手加热 DS18B20 模拟升温），观察协调器是否接收数据并打开风扇，串口是否显示“Fan ON”，验证完整功能。

### (3) 总结 (15 分钟)

工程创建流程：复制模板→配置参数→裁剪文件→编写代码→编译烧录→验证；

核心要点：文件裁剪“保留核心依赖”、堆栈配置“按需设置（避免溢出/浪费）”、功能整合“模块化拆分（采集/收发/控制）”。

### (4) 作业

基础题：在自定义工程中添加“风扇关闭”逻辑（温度 < 28°C 时 P1.3=0），重新编译验证；

拓展题：基于自定义工程，添加“多终端节点管理”（协调器接收 3 个终端温度，显示最高温度并控制风扇），提交工程文件与测试视频。

### (5) 反思

学生易在“堆栈大小配置”上出错（设置过小导致内存溢出，过大浪费资源），下次课可提供“堆栈配置参考表”（如终端节点 APP\_MAX\_MSGS=5，协调器=8）；部分学生文件裁剪后忘记删除冗余头文件引用（如保留 SampleApp.h 引用），需在编译后重点检查“未定义标识符”报错，引导学生定位问题。

## 第十七周：路由器多终端点播通信（12月29日 - 1月2日）

### 1. 教学课题和内容

**课题：**WSN 路由器节点配置与多终端点播转发

**内容：**①路由器节点的作用（转发多终端节点与协调器的点播数据，扩展通信距离）；②Zstack 路由器节点配置（DEVICE\_TYPE=ROUTER，使能路由功能）；③路由器转发原理（维护路由表，根据目标地址选择转发路径）；④实验：路由器多终端点播（搭建“终端1→路由器→协调器”“终端2→路由器→协调器”的点播路径，实现终端1发送“LED\_ON”→路由器转发→协调器点亮LED；终端2发送“LED\_OFF”→路由器转发→协调器熄灭LED）。

### 2. 教学课时

总学时 2 学时（理论 1 学时 + 实验 1 学时）

### 3. 学生已有知识分析

**已有基础：**学生已掌握终端与协调器的点播通信，熟悉 Zstack 节点配置（DEVICE\_TYPE）；了解“路由器用于扩展通信距离”，对“路由表存储节点路径”有初步认知。

**知识缺口：**不熟悉路由器节点的特殊配置（如使能路由功能的宏定义 NWK\_ROUTER\_CAPACITY=1）；对“路由器转发的延迟与路径选择”（自动选择最短路径）陌生；缺乏“多终端通过路由器点播的冲突处理”经验。

### 4. 教学目标

**知识目标：**理解路由器节点的作用与转发原理，掌握路由器节点配置方法，明确多终端通过路由器点播的路径逻辑。

**能力目标：**能配置路由器节点，能搭建多终端 - 路由器 - 协调器的点播路径，能验证转发功能与冲突处理。

**素养目标：**培养“网络拓扑扩展”的全局思维，通过路由转发培养“高效数据传输”的工程意识。

### 5. 教学重点

路由器节点配置（DEVICE\_TYPE=ROUTER，使能路由功能）；  
多终端 - 路由器 - 协调器的点播路径搭建与转发验证。

### 6. 教学难点

路由器路由表的维护（自动更新节点路径，如终端1移动后路由表更新）；  
多终端同时点播的转发冲突（路由器通过队列缓存数据，避免同时转发）。

### 7. 课程思政

展示“路由器在大型 WSN 中的应用”（如智慧园区中，路由器连接不同楼宇的终端节点，实现远距离通信），说明“网络扩展对物联网规模化应用的重要性”，培养学生的产业全局观；  
类比“路由器的转发功能”与“生活中‘驿站传递信件’”，引导学生理解“中间节点的桥梁作用”，树立“协作共赢”的意识。

## 8. 教学方法

理论教学：讲授法（结合路由器拓扑图与路由表）、动画演示法（播放路由转发动画）；  
实验教学：分组实操法（4 组分工：2 组终端、1 组路由器、1 组协调器）、故障排查法（模拟终端远离协调器，验证路由器转发是否生效）。

## 9. 教学程序

### (1) 导入 (15 分钟)

场景导入：“终端节点与协调器距离 50 米，超出无线通信范围（约 30 米），如何让终端数据仍能传输到协调器？”引出路由器节点；

演示导入：展示“终端→协调器（无路由，通信失败）vs 终端→路由器→协调器（有路由，通信成功）”的对比，提问“路由器如何知道数据该转发到哪里？需要什么特殊配置？”

### (2) 教授 (45 分钟)

理论讲解（20 分钟）：

路由器节点作用与原理：①作用：转发终端与协调器的点播数据，扩展 WSN 通信距离（单跳 30 米，多跳可扩展至百米）；②转发原理：路由器维护路由表（存储终端地址与下一跳地址），接收终端数据后，查询路由表，将数据转发到下一跳（如终端 1→路由器→协调器，下一跳为协调器地址 0x0000）；

路由器配置：①DEVICE\_TYPE=ROUTER（在 f8wConfig.cfg 中设置）；②使能路由功能：宏定义 NWK\_ROUTER\_CAPACITY=1（允许路由器存储路由表）；③其他配置：与终端 / 协调器一致（端点号 0x01、簇 ID 0x0001、波特率 9600bps）；

多终端点播路径：①终端 1（地址 0x0007）、终端 2（0x0008）→路由器（0x0009）→协调器（0x0000）；②路由表更新：终端上电后自动发送“路由请求”，路由器接收后更新路由表，记录终端地址与自身为下一跳。

实验操作（25 分钟）：

节点配置（10 分钟）：①路由器：DEVICE\_TYPE=ROUTER，NWK\_ROUTER\_CAPACITY=1，地址 0x0009；②终端 1：地址 0x0007，目标地址设为协调器 0x0000（无需指定路由器，协议栈自动发现路由）；③终端 2：地址 0x0008，指令“LED\_OFF”；④协调器：地址 0x0000，LED 接 P1.0；

搭建路径与验证（10 分钟）：①按“终端 1 - 路由器 - 协调器”顺序放置（终端 1 与协调器距离 50 米，路由器在中间）；②终端 1 发送“LED\_ON”，观察协调器 LED 是否点亮（验证转发生效）；③终端 2 发送“LED\_OFF”，观察 LED 是否熄灭；

故障模拟（5 分钟）：①移除路由器，终端 1 发送指令，观察协调器是否接收失败；②重新接入路由器，验证通信恢复，确认路由器的转发作用。

### (3) 总结 (10 分钟)

转发流程梳理：终端发送点播指令→路由器接收并查询路由表→转发到协调器→协调器执行动作；

核心要点：路由器配置（ROUTER 类型 + 路由使能）、路由表自动维护（无需手动配置路径）、多终端冲突（路由器自动缓存排队转发）。

## (4) 作业

基础题：添加终端 3（地址 0x000A），验证路由器是否能同时转发 3 个终端的点播指令；  
拓展题：模拟路由器故障（断电），观察终端是否能重新发现新路由（如另一个备用路由器），提交测试报告。

## (5) 反思

学生易在“路由器路由使能宏定义”上遗漏（忘记设置 `NWK_ROUTER_CAPACITY=1`），导致路由表无法更新，下次课需在配置步骤中重点标注；  
部分学生对“路由表自动发现”理解困难，可通过 Z-Tool 读取路由器路由表，直观展示终端地址与下一跳的对应关系。

# 第十八周：课程总结及考核（1月5日 - 1月9日）

## 1. 教学课题和内容

课题：《无线网络应用》课程知识整合与综合实验考核

内容：①课程知识体系梳理（WSN 基础→Zstack 协议栈→CC2530 硬件→无线通信（单播 / 组播 / 点播 / 透传）→传感器采集→多设备联动）；②综合实验考核任务（分组完成“WSN 温湿度监测与联动控制系统”：3 个终端（DHT11/DS18B20 采集）→1 个路由器→1 个协调器（串口显示 + LED 报警 + 继电器控制风扇），实现“温度 > 30°C→风扇开 + 红灯亮，湿度 < 40%→黄灯亮”）；③考核标准（功能完整性 40%、代码规范性 30%、故障排查 20%、团队协作 10%）；④考核总结与反馈（点评共性问题，梳理学习建议）。

## 2. 教学课时

总学时 4 学时（理论 2 学时 + 实验 2 学时）

## 3. 学生已有知识分析

已有基础：学生已掌握课程核心知识点（WSN、Zstack、CC2530、传感器、无线通信、联动控制），具备独立完成单一模块功能的能力；了解“综合实验需整合多模块”，对“团队协作分工”有初步认知。

知识缺口：不熟悉“多模块整合的优先级规划”（如先搭建通信链路，再添加采集与控制）；对“综合实验的故障排查逻辑”（从通信→采集→控制逐步排查）陌生；缺乏“团队协作中的任务分配与进度管理”经验。

## 4. 教学目标

知识目标：构建完整的课程知识体系，明确各模块的逻辑关联，掌握综合实验的设计与实现方法。

能力目标：能团队协作完成综合实验，能按考核标准排查故障，能规范编写代码与提交报告。

素养目标：培养“系统整合与团队协作”的综合能力，通过考核反思培养“查漏补缺、持续提升”的学习习惯。

## 5. 教学重点

课程知识体系梳理（模块关联：采集→通信→控制→联动）；  
综合实验考核任务的功能实现与故障排查。

## 6. 教学难点

多模块整合的优先级规划（如先确保路由器转发正常，再添加传感器采集）；  
团队协作中的任务同步（如终端组与协调器组需统一指令格式）。

## 7. 课程思政

结合“综合实验考核”，强调“团队协作的重要性”（如采集组与通信组需密切配合，避免指令格式不统一），引导学生树立“分工协作、优势互补”的团队意识；  
总结课程学习历程，鼓励学生“将知识转化为解决实际问题的能力”（如 WSN 系统可应用于智慧农业、工业监测），培养学生“学以致用、服务产业”的职业追求。

## 8. 教学方法

理论教学：思维导图法（绘制课程知识体系图）、考核标准讲解法（明确评分细则）；  
实验教学：团队考核法（4-5 人一组，分工完成任务）、现场点评法（教师巡视各组，实时反馈问题）。

## 9. 教学程序

### (1) 导入 (20 分钟)

回顾导入：“从第一周的 WSN 概述到第十七周的路由器转发，我们学习了哪些核心模块？它们如何共同构成一个完整的 WSN 系统？”引出知识整合；  
考核布置：讲解综合实验考核任务与标准，明确“2 学时团队协作完成实验，1 学时演示与答辩，1 学时总结反馈”的流程。

### (2) 教授与考核 (100 分钟)

课程总结（30 分钟）：

知识体系梳理：①核心基础：WSN 定义与 Zstack 协议栈（OSAL 事件、AF 无线通信）；②硬件操作：CC2530 GPIO / 串口 / 定时器、传感器（DHT11/DS18B20）、执行器（LED / 继电器）；③通信方式：单播（终端→协调器）、组播（多终端→协调器）、点播（终端→终端）、透传（电脑→电脑）、路由转发（多跳扩展）；④系统联动：采集→通信→控制→报警（如温度高→风扇开+红灯亮）；

模块关联图：以“WSN 系统”为核心，分支为“感知层（传感器）→网络层（无线通信 / 路由）→应用层（联动控制）”，标注各模块的依赖关系（如联动控制依赖采集与通信）。

综合实验考核（70 分钟）：

团队分工（10 分钟）：各组内部分工（如 1 人负责路由器配置、2 人负责终端采集、2 人负责协调器控制），确定任务节点（15 分钟完成通信链路搭建，30 分钟完成采集与控制，15 分钟调试）；

实验实施（40 分钟）：①搭建拓扑：3 个终端→1 个路由器→1 个协调器；②功能实现：终端采集温湿度并发送（添加“节点号，H,T”格式），路由器转发，协调器解析数据（温度 > 30°C→P1.3=1 开风扇 + P1.1=0 亮红灯，湿度 < 40%→P1.2=0 亮黄灯），串口显示所有终端数据；③故障排查：教师巡视，针对“通信失败”（检查地址 / 簇 ID）、“采集异常”（检查传感器接线）、“控制无效”（检查 GPIO 配置）等问题提供提示；

演示与答辩（20分钟）：各组依次演示功能（如加热终端1传感器，观察协调器风扇与红灯），回答教师提问（如“路由器未工作时如何排查？”“湿度报警阈值如何修改？”），教师按标准评分。

### **(3) 总结（15分钟）**

考核总结：①共性问题：指令格式不统一（终端与协调器字段顺序不一致）、路由配置遗漏（未使能 `NWK_ROUTER_CAPACITY`）、代码无注释（难以排查）；②优秀案例：某组添加“数据丢失重传”功能，某组实现“协调器向终端发送阈值修改指令”；

学习建议：①巩固基础：重点复习 Zstack 事件处理与无线通信；②拓展学习：学习 LoRa/Wi-Fi 物联网通信技术，对比 WSN 的差异；③实践应用：尝试将 WSN 系统应用于校园环境监测（如教室温湿度）。

### **(4) 作业**

总结报告：撰写“综合实验考核总结报告”，含团队分工、功能实现步骤、故障排查过程、学习反思（不足与改进方向）；

拓展任务：调研“WSN 与 LoRa 的技术差异”，撰写 1000 字对比报告，为后续课程学习铺垫。

### **(5) 反思**

考核中发现部分团队分工不明确（如多人同时修改协调器代码），下次课可提供“团队分工模板”（明确每人职责与时间节点）；

部分学生对“多模块整合的优先级”把握不当（先做控制再做通信，导致后续返工），需在考核前强调“先保障通信，再添加其他功能”的逻辑。