**福建信息职业技术学院教案**

第 11 号

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | 嵌入式操作系统及应用 | | 授课日期 |  |
| 班 级 | 物联网2411|物联网2412 | | 课堂类型 | 一体化 |
| 教 材 | OpenHarmony嵌入式系统原理与应用——基于RK2206芯片 | | | |
| 章节名称 | 基础开发  [标准PWM控制](https://gitee.com/Lockzhiner-Electronics/lockzhiner-rk2206-openharmony3.0lts/blob/master/vendor/lockzhiner/rk2206/samples/a8_hal_pwm/README_zh.md) | | | |
| 目的要求 | - 理解PWM（脉冲宽度调制）基本原理（频率、占空比定义）； - 掌握OpenHarmony PWM核心API（IoTPwmInit()初始化、IoTPwmStart()输出、IoTPwmStop()停止）的参数与功能； - 明确小凌派-RK2206 PWM端口与GPIO的对应关系（如端口0对应GPIO\_PB4）。 | | | |
| 学情分析 | 1. 基础：已掌握任务创建、编译配置流程，具备C语言循环基础，但对硬件接口（PWM端口与GPIO对应）、PWM参数（占空比、频率）认知薄弱； 2. 难点：易混淆PWM端口号与GPIO的对应关系，忽略IoTPwmStart()占空比范围（1~99）； 3. 需求：需通过硬件端口图解、参数错误演示降低理解难度。 | | | |
| 重 难 点  分 析 | 1. 重点： - API应用：IoTPwmInit(port)（port为0~10）、IoTPwmStart(port, 50, 1000)（占空比50、频率1000Hz）； - 编译配置：修改vendor/lockzhiner/rk2206/sample/BUILD.gn添加./a8\_hal\_pwm:hal\_pwm\_example，Makefile添加-lhal\_pwm\_example； - 结果验证：串口观察“每隔5秒切换PWM端口，打印PWM(x) start/end”。 2. 难点： - 硬件对应：PWM端口与GPIO的匹配（如端口3对应GPIO\_PC0）； - 参数容错：IoTPwmStart()占空比超范围（如0或100）导致函数失败的处理。 | | | |
| 信息化应用方法 | 网络教学平台、视频、开发板，开展现场教学，通过项目任务驱动进行混合式教学； | | | |
| **思政元素**  **融合设计** | 思政元素 | 融入方式 | | |
| 民族自豪感、  科技报国情怀 | 1. 讲解小凌派-RK2206 PWM在国产智能家居（如LED调光台灯）、工业电机控制中的应用，说明国产开发板对硬件接口的本土化适配优势； 2. 对比国外开发板PWM接口的封闭性，强调鸿蒙系统“硬件接口开源、工具链自主”的价值，引导学生认同国产嵌入式生态。 | | |
| 作业布置 | |  | | --- | | 1. 实操任务： - 修改代码：将IoTPwmStart()占空比改为30、70，观察串口日志，提交修改后代码与日志截图； - 新增功能：在端口5停止后，延迟2秒再启动端口6，验证PWM端口切换的灵活性。 2. 预习任务： - 阅读参考资料中“鸿蒙硬件GPIO控制”章节，了解IoTGpioInit()基本功能，为下次课程铺垫。 | | | | |
| 参考资料 | 1.OpenHarmony嵌入式系统原理与应用——基于RK2206芯片  2.https://gitee.com/Lockzhiner-Electronics/lockzhiner-rk2206-openharmony3.0lts.git | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 教学  环节 | 教学  内容 | 教师活动 | 学生活动 | 设计意图 |
| 课前 | 预习（30分钟） | 1. 上传预习资料： - PWM原理动画视频（占空比、频率直观演示）； - 小凌派-RK2206 PWM端口与GPIO对应表； 2. 发布任务：标注IoTPwmStart()三个参数含义，记录“占空比50代表什么”的疑问。 | 1. 观看视频，记录端口-GPIO对应关系； 2. 提交预习疑问，在平台互动。 | 提前铺垫PWM基础，降低课中硬件认知难度。 |
| 课中 | 课程介绍（5min） | 1. 明确目标：掌握PWM API与硬件控制，理解国产开发板优势； 2. 串联逻辑：从“PWM应用场景”到“代码实现”，融入思政目标。 | 1. 记录核心目标； 2. 提问“PWM还能控制什么设备”。 | 清晰学习方向，激发硬件控制兴趣。 |
| 任务导入  （10min） | 1. 演示实验效果：串口展示“PWM(0) start→5秒后end，切换PWM(1)”的日志； 2. 提问：“如何实现每隔5秒切换PWM端口？端口对应的GPIO有什么用？” | 1. 观察日志规律； 2. 分组讨论，初步梳理“初始化→启动→停止”流程。 | 用直观效果激发探索欲，聚焦核心任务。 |
| 知识储备  （15min） | 1. 讲解核心知识： - PWM原理：频率（每秒脉冲数）、占空比（高电平占比）； - API解析：IoTPwmInit()（端口初始化）、IoTPwmStart()（参数范围）； - 硬件对应：展示开发板PWM端口实物图解； 2. 思政融入：穿插国产PWM应用案例（如智能调光）。 | 1. 绘制PWM波形图； 2. 记录端口-GPIO对应关系。 | 夯实理论，结合硬件实物降低抽象难度 |
| 任务导入  （5min） | 1. 明确实操任务：创建a8\_hal\_pwm文件夹，编写循环控制PWM端口的代码； 2. 强调易错点：端口号范围（010）、占空比（199）。 | 1. 记录实操步骤； 2. 标注易错点。 | 明确任务边界，减少操作失误。 |
| 知识储备  （10min） | 1. 补充细节： - 代码结构：pwm\_example()创建任务，hal\_pw\_thread()实现PWM循环控制； - 配置语法：BUILD.gn与Makefile的库名、路径匹配； 2. 错误演示：占空比设为0，展示IoTPwmStart()失败日志。 | 1. 记录代码模板； 2. 练习根据日志排查参数错误。 | 补充实操细节，提升问题解决能力。 |
|  | 任务实施  （40min） | 1. 分步演示+指导： ① 创建a8\_hal\_pwm文件夹； ② 编写代码（含任务创建、PWM循环控制）； ③ 修改BUILD.gn和Makefile； ④ 烧写验证； 2. 重点帮扶：解决端口对应错误、配置路径问题。 | 1. 跟随操作，每步自查； 2. 遇错先排查参数/路径，再求助； 3. 成功后拍照记录日志。 | 通过实操突破重点，针对性解决难点。 |
|  | 任务总结  （5min） | 1. 梳理流程：代码→配置→烧写→验证； 2. 强化重点：端口-GPIO对应、参数范围； 3. 思政升华：国产开发板硬件控制的自主优势。 | 1. 补充流程笔记； 2. 分享实操心得。 | 巩固知识，深化思政认知。 |
| 作业 |  | 1. 在学习通发布作业：明确代码命名、截图要求； 2. 提示下次课重点：GPIO控制。 | 1. 记录作业要求； 2. 规划完成时间。 | 验收成果，铺垫后续课程。 |
| 课后 |  | |  | | --- | | 1. 上传PPT、实操视频、错误排查手册； 2. 24小时内回复问题，汇总高频错误。 | | 1. 下载资源复盘； 2. 提交疑问，查看解答。 | 提供复习支持，帮助查漏补缺。 |