**福建信息职业技术学院教案**

第 21 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 | 基础案例  [看门狗控制](https://gitee.com/Lockzhiner-Electronics/lockzhiner-rk2206-openharmony3.0lts/blob/master/vendor/lockzhiner/rk2206/samples/b12_watchdog/README_zh.md) | | | |
| 目的要求 | 1. 知识目标： - 理解看门狗（Watchdog）工作原理（监控系统运行，超时未喂狗则重启）； - 掌握核心API（LzWatchdogInit()初始化、LzWatchdogSetTimeout()设超时、LzWatchdogKeepAlive()喂狗）功能； - 明确超时时间档位限定（如1.3981013\*(2^n)秒）。 2. 技能目标： - 能创建b12\_watchdog文件夹，编写看门狗控制代码（初始化+条件喂狗）； - 独立修改BUILD.gn和Makefile完成编译，烧写后通过串口验证“喂狗-重启”效果。 | | | |
| 学情分析 | 1. 基础：已掌握GPIO、UDP等外设开发，具备C语言循环逻辑基础，但对“喂狗机制”“超时档位”认知薄弱； 2. 难点：易忽略超时时间的档位限定（非任意值），混淆“喂狗时机”与“重启触发”逻辑； 3. 需求：需通过代码错误演示（如不喂狗导致重启）、档位计算示例降低难度。 | | | |
| 重 难 点  分 析 | 1. 重点： - API应用：LzWatchdogInit()初始化、LzWatchdogSetTimeout(20)设超时（实际约22.37秒）、LzWatchdogKeepAlive()喂狗； - 编译配置：修改vendor/lockzhiner/rk2206/sample/BUILD.gn添加./b12\_watchdog:watchdog\_example，Makefile添加-lwatchdog\_example； - 结果验证：串口观察“current1-10喂狗（freedog）、11后不喂狗（not freedog），最终系统重启”。 2. 难点： - 超时档位：理解1.3981013\*(2^n)的计算逻辑（如n=4对应22.37秒）； - 重启触发：未在超时前喂狗导致系统重启的时序逻辑。 | | | |
| 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看门狗在国产工业控制器、智能网关中的应用，说明其对设备长期稳定运行的支撑作用，体现国产开发板的可靠性优势； 2. 对比国外开发板看门狗驱动的封闭性，强调鸿蒙系统“看门狗接口开源、底层可控”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LzWatchdogSetTimeout()参数改为15，计算实际超时时间（约1.3981013\*(2^3)=11.18秒），验证“current1-8喂狗、9后重启”效果，提交代码与串口日志截图； - 拓展功能：添加“按键触发喂狗”逻辑（GPIO按键按下时执行LzWatchdogKeepAlive()）。 2. 预习任务： - 阅读参考资料中“鸿蒙系统综合案例（多外设联动）”章节，了解看门狗与GPIO、PWM的协同应用思路。 | | | |
| 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. 在学习平台上传预习资料： - 看门狗“喂狗-重启”动画视频（演示超时触发逻辑）； - 核心API参数图解、超时档位计算示例（如n=4对应22.37秒）； 2. 发布预习任务：标注LzWatchdogSetTimeout()的参数含义，记录“为什么超时时间不是任意值”的疑问。 | 1. 观看视频，计算并记录不同n对应的超时时间； 2. 提交预习疑问，在平台互动交流。 | 提前铺垫看门狗核心逻辑与档位计算，降低课中抽象概念理解难度。 |
| 课中 | 课程介绍（5min） | 1. 明确本节课核心目标：掌握看门狗API与“喂狗”逻辑，理解国产开发板稳定性保障机制； 2. 串联课程逻辑：从“设备稳定运行需求”到“看门狗解决方案”，融入思政目标。 | 1. 记录核心目标，标注“超时档位”“喂狗时机”两个重点； 2. 提问“工业设备为什么需要看门狗”。 | 清晰学习方向，激发对设备稳定性保障的探索兴趣，关联思政目标。 |
| 任务导入  （10min） | 1. 演示两组对比实验： - 实验1（正常喂狗）：串口打印“current1-10 freedog”，系统稳定运行； - 实验2（不喂狗）：current11后打印“not freedog”，约22秒后系统重启（串口显示“entering kernel init...”）； 2. 提问引导：“为什么实验2会重启？喂狗的核心作用是什么？” | 1. 观察两组实验的串口日志差异，记录重启时序； 2. 分组讨论问题，初步梳理“初始化→喂狗→超时判断”流程。 | 用直观对比实验呈现看门狗核心价值，通过问题聚焦“喂狗机制”与“重启触发”关键点。 |
| 知识储备  （15min） | 1. 分层讲解核心知识： - 看门狗本质：类比“工厂安全员”（定时检查，超时则触发应急措施），核心是“超时监控”与“故障恢复”； - API解析： - LzWatchdogInit()：初始化看门狗模块，开启监控； - LzWatchdogSetTimeout(UINT32 timeout)：设置超时参数（需符合档位，如timeout=20对应n=4）； - LzWatchdogKeepAlive()：喂狗操作（重置超时计时，避免重启）； - 超时计算：推导1.3981013\*(2^n)的逻辑，结合示例验证（如timeout=20对应22.37秒）； 2. 思政融入：穿插国产工业设备（搭载鸿蒙看门狗）的稳定运行案例，对比国外系统驱动依赖风险。 | 1. 绘制“看门狗工作流程图”，记录API参数与档位计算； 2. 举手提问“如何确定n的取值”，标记“喂狗时机判断方法”。 | 用生活化类比降低抽象难度，结合国产案例强化思政认知，夯实理论基础。 |
| 任务导入  （5min） | 1. 承接知识储备，明确本次实操任务： - 在vendor/lockzhiner/rk2206/sample下创建b12\_watchdog文件夹； - 编写看门狗控制代码（含初始化、条件喂狗），修改配置文件，验证“喂狗-重启”效果； 2. 强调实操关键点： - 代码需包含“current1-10喂狗、11后不喂狗”的条件逻辑； - BUILD.gn与Makefile的库名、路径必须匹配。 | 1. 记录实操步骤框架（创建文件夹→写代码→改配置→烧写验证）； 2. 标注“文件夹名是b12\_watchdog”“库名是watchdog\_example”两个易错点。 | 明确任务边界与实操重点，减少代码与配置失误，为后续实操铺垫。 |
| 知识储备  （10min） | 1. 补充实操细节知识： - 代码结构：watchdog\_process()实现初始化→循环喂狗（current≤10时调用LzWatchdogKeepAlive()）； - 配置文件修改语法： - BUILD.gn：在features中添加./b12\_watchdog:watchdog\_example； - Makefile：在app\_LIBS后添加-lwatchdog\_example； - 错误预判：演示“超时参数设为50（超出档位）”的编译报错，教学生根据日志排查档位问题。 | 1. 记录代码模板（如循环喂狗的条件判断逻辑）； 2. 模拟“配置文件库名写错”的错误，练习通过日志定位问题。 | 补充实操细节与错误排查方法，提升学生问题解决能力，确保实操顺利推进。 |
|  | 任务实施  （40min） | 1. 分步演示+巡视指导： ① 终端创建b12\_watchdog文件夹（mkdir -p b12\_watchdog）； ② 编写watchdog\_example.c： - 包含头文件#include "lz\_watchdog.h"； - 实现watchdog\_process()（初始化、循环喂狗、打印current值）； ③ 修改BUILD.gn和Makefile：按语法添加编译目标与库名； ④ 烧写程序，打开串口软件（波特率115200），观察“喂狗-重启”日志； 2. 重点帮扶：针对“超时档位计算错误”“配置文件路径错”等问题，一对一指导排查（如通过printf打印实际超时时间）。 | 1. 跟随演示逐步操作，每完成一步自我检查（如核对BUILD.gn路径）； 2. 遇到错误先对照预习资料中的档位表排查，无法解决则举手求助； 3. 成功观察到重启日志后，拍照记录实验结果。 | 通过“分步演示+针对性指导”突破重点，解决“档位计算”“配置关联”难点，强化动手能力。 |
|  | 任务总结  （5min） | 1. 流程梳理：用思维导图回顾“代码编写→配置修改→烧写→喂狗验证”全步骤； 2. 重难点强化： - 核心逻辑：喂狗是“重置超时计时”，超时未喂狗则触发重启； - 关键细节：超时参数需符合1.3981013\*(2^n)档位，配置文件库名与路径必须匹配； 3. 思政升华：展示国产工业网关（搭载鸿蒙看门狗）的稳定运行案例，强调自主可控技术对设备可靠性的支撑作用。 | 1. 补充思维导图细节，标记“档位计算”“配置关联”两个易错点； 2. 分享实操心得（如“原来参数20对应的实际超时是22秒”）。 | 巩固知识点与实操流程，结合实际应用深化思政认知，形成“理论-实操-应用”的完整学习闭环。 |
| 作业 |  | 1. 在学习通发布作业： - 实操作业要求：代码命名格式为“watchdog\_example\_姓名.c”，日志截图需标注“喂狗阶段”“重启阶段”； - 预习作业要求：简述看门狗与GPIO、PWM联动的思路（不少于150字）； 2. 提示下次课重点：鸿蒙多外设联动综合案例。 | 1. 记录作业要求，规划完成时间（实操1.5小时+预习30分钟）； 2. 课后回顾本节课代码，为修改超时参数的实操任务做准备。 | 通过作业验收实操成果，强化看门狗逻辑理解；预习多外设联动，为后续综合案例铺垫，保持知识连贯性。 |
| 课后 |  | 1. 上传教学资源： - 本节课PPT（含看门狗原理图解、API参数表）； - 实操步骤视频（含配置文件修改、串口日志解读）； - 错误排查手册（常见错误如档位错、库名错及解决方法）； 2. 开放答疑通道： - 24小时内回复学习平台的代码问题； - 汇总“超时档位计算错误”“配置文件路径错”等高频问题，下次课开篇讲解。 | 1. 登录学习平台下载资源，对照视频复盘实操过程； 2. 完成作业时遇到问题及时提问，查看同学的疑问及解答，补充自身知识漏洞。 | 提供课后复习支持，及时解决学习问题；通过高频错误汇总优化后续教学，帮助学生查漏补缺，巩固学习效果。 |