**福建信息职业技术学院教案**

第 16 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  [UART打印](https://gitee.com/Lockzhiner-Electronics/lockzhiner-rk2206-openharmony3.0lts/blob/master/vendor/lockzhiner/rk2206/samples/b6_uart0/README_zh.md) [UART2收发](https://gitee.com/Lockzhiner-Electronics/lockzhiner-rk2206-openharmony3.0lts/blob/master/vendor/lockzhiner/rk2206/samples/b6_uart2/README_zh.md) | | | |
| 目的要求 | - 理解UART2工作原理（基于异步通信，GPIO0\_PB2=RX、PB3=TX）； - 掌握核心API（LzUartInit()初始化、LzUartRead()读、LzUartWrite()写）功能； - 明确串口参数（波特率115200、8位数据位、1位停止位）。 | | | |
| 学情分析 | 1. 基础：已掌握I2C/OLED外设开发，具备C语言数组处理基础，但对UART异步收发、FIFO缓存认知薄弱； 2. 难点：易混淆串口引脚（RX/TX接反），忽略FIFO接收任务的低延时要求； 3. 需求：需通过引脚接线图、收发错误演示降低难度。 | | | |
| 重 难 点  分 析 | 1. 重点： - API应用：LzUartInit()（配置波特率115200、引脚复用）、LzUartWrite()（异步发送）、LzUartRead()（FIFO接收）； - 编译配置：修改vendor/lockzhiner/rk2206/sample/BUILD.gn添加./b6\_uart2:uart2\_example，Makefile添加-luart2\_example； - 结果验证：串口助手接收开发板发送数据，向开发板发送数据并在串口日志查看。 2. 难点： - 硬件接线：RX（PB2）接串口模块TX，TX（PB3）接串口模块RX； - 接收任务：FIFO缓存数据的逻辑，避免接收任务延时导致数据丢失。 | | | |
| 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 | 通过国产软件的崛起及应用事例的讲解，树立学生的民族自豪感和科技报国情怀 | | |
| 作业布置 | 1. 实操任务： - 修改代码：实现“开发板接收串口助手数据后，回发相同数据”的功能，提交代码与串口助手截图； - 优化功能：调整波特率为9600，验证不同波特率下的收发稳定性。 2. 预习任务： - 阅读参考资料中“鸿蒙SPI触摸屏控制”章节，了解IoTSpiWrite()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： - UART异步通信动画视频（RX/TX数据交互演示）； - 小凌派-RK2206 UART2引脚（PB2/RX、PB3/TX）接线图； 2. 发布任务：标注LzUartInit()的串口参数含义，记录“FIFO缓存作用”的疑问。 | 1. 观看视频，记录引脚接线规则； 2. 提交预习疑问，在平台互动。 | 提前铺垫UART硬件与缓存基础，降低课中接线与代码理解难度。 |
| 课中 | 课程介绍（5min） | 1. 明确目标：掌握UART2 API与收发控制，理解国产串口通信优势； 2. 串联逻辑：从“串口通信场景”到“代码实现”，融入思政目标。 | 1. 记录核心目标； 2. 提问“UART还能用于哪些设备通信”。 | 清晰学习方向，激发串口通信开发兴趣。 |
| 任务导入  （10min） | 1. 演示实验效果： - 开发板通过UART2向串口助手发送“Hello UART2”； - 串口助手向开发板发送数据，开发板串口日志显示接收数据； 2. 提问：“如何避免串口接收数据丢失？RX和TX为何不能接反？” | 1. 观察串口助手与开发板日志； 2. 分组讨论问题，初步梳理“初始化→发送→FIFO接收”流程。 | 用直观通信效果激发探索欲，聚焦UART接线与缓存核心点。 |
| 知识储备  （15min） | 1. 讲解核心知识： - UART原理：异步通信，波特率115200，RX/TX交叉接线； - API解析：LzUartInit()（参数配置、引脚复用）、LzUartWrite()（异步非阻塞）、LzUartRead()（FIFO缓存）； - 接收任务：创建低延时任务，避免数据丢失； 2. 思政融入：穿插国产工业串口设备（鸿蒙驱动）案例，对比国外技术依赖。 | 1. 记录API参数与接线规则； 2. 标注FIFO接收任务的关键代码。 | 夯实理论，结合硬件接线降低抽象难度。 |
| 任务导入  （5min） | 1. 明确实操任务：创建b6\_uart2文件夹，编写UART2收发代码； 2. 强调易错点：RX/TX交叉接线、FIFO接收任务无多余延时。 | 1. 记录实操步骤； 2. 标注“接线反会导致无数据”。 | 明确任务边界，减少接线与代码失误 |
| 知识储备  （10min） | 1. 补充细节： - 代码结构：uart2\_example()初始化UART+创建接收任务，uart2\_recv\_process()实现FIFO接收； - 配置语法：BUILD.gn与Makefile的库名、路径匹配； 2. 错误演示：接线反导致无数据，展示串口日志“recv null”。 | 1. 记录代码模板； 2. 练习根据日志排查接线错误。 | 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b6\_uart2文件夹； ② 编写代码（UART初始化、发送数据、FIFO接收任务）； ③ 指导接线（PB2→串口TX，PB3→串口RX）； ④ 修改BUILD.gn和Makefile，烧写验证； 2. 重点帮扶：解决接线错误、FIFO任务延时问题。 | 1. 跟随操作，每步自查（接线、代码）； 2. 遇错先排查接线/延时，再求助； 3. 成功后用串口助手收发数据，截图记录。 | 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→接线→配置→烧写→通信验证； 2. 强化重点：RX/TX交叉接线、FIFO缓存逻辑； 3. 思政升华：国产UART在工业通信中的自主应用价值。 | 1. 补充流程笔记； 2. 分享实操心得（如“接线反后无数据，重新接线就好”）。 | 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、串口助手截图要求； 2. 提示下次课重点：SPI触摸屏控制。 | | 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（如接线反、任务延时）。 | 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。 |