**NK2024春·02#深度课设需求v5.0**

**NK2024·02#深度课设需求建议V5.0**

**（本文档各组需根据本组具体情况重新填写）**

课设主题：车联一 本主题小组数：2

第02#组所在房间：205

组 长：巩岱松

需求员：郭子涵

周志员：武宇航

优裁员：王晶

展示员：金莫迪

组长助理：何叶

**目录**

[一、深度课设需求整体说明 3](#_Toc28458)

[二、授设同步递进·版本要求 9](#_Toc21515)

[三、量化指标·边测设置 24](#_Toc18700)

[四、软硬结合三算·设计痕迹 25](#_Toc15943)

[五、两套UI·内外双构 30](#_Toc25676)

[六、本组化定制痕迹 42](#_Toc6603)

[七、基础底座与主题特色痕迹 48](#_Toc6296)

[八、可嵌入系统整体性痕迹（选做） 50](#_Toc15491)

[九、课程与课设关联痕迹 53](#_Toc13627)

[十、gitlab·issue与认知OKR 56](#_Toc3417)

[十一、展示分享与反省 57](#_Toc12057)

[附件一【问题递进闭环·样例建议模板】浅度课设·双UI解耦群组点名提升 62](#_Toc14524)

[附件二 课程三栏认知OKR建议 72](#_Toc18330)

[目标： 72](#_Toc28253)

[通过浅度到深度课设的递进实践，掌握嵌入式系统开发全流程技能，实现从代码编写到系统优化的能力跃升，同时强化团队协作与问题解决意识。 72](#_Toc15338)

[O1：掌握嵌入式基础开发流程与核心技能 72](#_Toc6645)

[O2：提升系统设计与优化能力 73](#_Toc18227)

[O3：强化团队协作与问题解决能力 73](#_Toc22281)

[O4：完成文档撰写与成果展示 73](#_Toc22468)

[附件三：学习认知节律 76](#_Toc8204)

|  |  |  |
| --- | --- | --- |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **深浅课设流畅对接**  **熟练运用五层结构，构造初步嵌入式系统** | **深度课设需求任务1·熟悉深度课设** | **展示** |
|  | 一、深度课设需求整体说明 |  |
| **操作发问**  如何通过拨码开关实现最简 / 正常 UI 模式切换？  **编程发问**  如何在 Java 代码中实现 AD 电位器采集温度值并驱动数码管显示？  **任务发问**  从单体运行到群驾模式，系统复杂度如何逐步提升？需解决哪些关键技术节点（如通信协议、冲突仲裁）？  **递进发问**  浅度课设的双 UI 点名功能如何迁移至车联场景的温控与组队需求？  **关联发问**  浅度课设的双 UI 点名功能如何迁移至车联场景的温控与组队需求？ | 1、什么是深度课设？  深度课设任务，是承接浅度课设Java群组点名任务，以主题对象分析对接·系统结构设计规划·环节模拟实现验证为目标，主要使用Java语言编写代码，在嵌入式教学实验箱Android系统平台上运行，在Stage2·深度课设阶段中分别开发纸联·练字笔迹、车联Ⅰ·E0-smart、车联Ⅱ·共享单车和体联·五时穿戴四个不同主题任务的嵌入式系统设计实践核心环节。  以本组的“车联I·E0-smart”智能系统为例，我们在系统设计中充分贯彻深度课设的理念与目标：  1. 在UI“点名”界面中，设计了十个模拟按钮与统一密码机制，用于成员身份识别与登录，剥离繁琐操作流程，突出核心功能，正是“除法则”在界面设计中“保留核心、舍弃冗余”的体现；  2.主界面采用上中下三区结构，分别展示身份信息、控制操作与车队状态反馈，提升用户操作的直观性与系统数据透明度；  3.在车队形成与运行控制中，系统支持基于位置差异的动态建队机制，根据不同区域的限速策略自动调节车速，队列成员通过逐步提速与队尾放缓进行协同配合，提升整体调度效率——这是“乘法则”在协同控制与系统集成中的体现；  4.引入风阻协同与能量共享机制，例如通过降低尾部风阻、共享电量等手段优化队列运行效率，同时结合车辆变姿能力，提高系统响应性与续航力；  5. 通过动画效果（如震动、旋转等）增强系统状态的可视化表达，提升模拟真实感；  6. 在“加法则”方面，我们设定电量阈值触发机制，在低电状态下自动渐进式减速，增强系统韧性与安全反馈；  7.“返回主界面”功能与车辆逐步停止控制机制体现了良好的容错性与人机交互逻辑；  8.针对老年用户群体，UI界面中特别加入\*\*按键放大、高亮蓝光提示\*\*等人性化设计，体现了课设对实际应用对象的人文关怀与适老化适配思路。  此外，本任务全过程依托GitLab版本控制与代码管理机制，确保代码质量可控、版本演化可溯，配合系统功耗监测、性能观测、日志记录与报告编写等配套工作，最终实现了一个结构清晰、逻辑完备、功能互联、可部署运行的嵌入式智能系统原型。  通过本学期的课设实验，全组同学不仅在Java与嵌入式平台上实现了完整的系统功能，更在需求抽象、功能裁剪、协同优化、人机交互等方面实现了理论与实践的深度融合。最终通过GitLab版本管理与阶段测试，逐步递进地完成一个可运行、可测量、可优化、具协作性的嵌入式系统雏形，实现从“点”到“面”的能力突破，真正做到了**“记得住、**  **说得出、内心渴望”，**这正是深度课设的真正价值所在。  2、课设实验箱面板深度课设使用硬件I/O图示意 | **结果展示**  组队成功时，LED 灯组显示 “01”（入队）或 “10”（头车），蜂鸣器播放提示音。  **问题展示**  休眠模式下电流仍达 0.8A，远超设计目标（≤0.5A），经查为传感器未完全断电。  **困难展示**  对 Android 系统底层驱动开发经验不足，导致 LED 灯位操作与触摸事件响应逻辑调试耗时较长。  **反省展示**  应在需求分析阶段提前明确硬件接口分配规则，避免后期冲突。  **合作共生展示**  硬件组设计 “变姿控制模块” 电路原理图，软件组基于此开发驱动代码，测试组编写边界值测试用例。 |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **深浅课设流畅对接**  **熟练运用五层结构，构造初步嵌入式系统** | **深度课设需求任务1·熟悉深度课设** | **展示** |
| **操作发问**  变姿控制（四轮正轮态→360 旋转轮态）的硬件触发逻辑如何设计？  **编程发问**  结伴运行逻辑中，如何通过线程同步确保多车状态实时同步  **任务发问**  三法则（除 / 乘 / 加）在车联系统优化中如何具体落地（如简化非核心功能、放大结伴激励策略）？  **递进发问**  初级 UI（数码管 + 拨码开关）到高级 UI（触摸屏 + 动画）的交互逻辑如何解耦与协同？  **关联发问**  如何在 Java 代码中实现 AD 电位器采集温度值并驱动数码管显示？ | 3、主要抽象逻辑概念定义  【字节位顺序定义】  LED灯：课设实验箱面板LED灯（左边两个）从左到右分别代表LED1-LED2 >> bit1-bit0  即LED灯从左到右分别对应一个2位二进制串的最高位到最低位（下同）  （某个LED灯灯亮表示该LED灯对应bit值为“1”，灯灭表示该bit值为“0”）  拨码开关：课设实验箱面板8位拨码开关，位置从“8”到“1”分别代表：  “8”-“1” >> bit7-bit0  （某一位开关拨动到“打开”状态表示该位开关对应bit值为“1”，“关闭”状态表示该bit值为“0”）   1. 项目的核心数据结构包括Car类（管理车速、电量、姿态）、CarTeam类（管理车队队列）、Location类（地理位置）、MapManager（提供地图坐标服务）。关键变量如carElectricity、carSpeed、runDistance直接影响状态变化与回滚触发。 2. 软看门狗逻辑定义：在嵌入式系统中，看门狗定时器（Watchdog Timer，简称WDT）是一种重要的硬件功能，用于检测和恢复系统故障。当程序运行异常或进入死循环时，看门狗定时器可以自动重启系统，确保设备能够恢复正常工作。Arduino微控制器也内置了看门狗定时器。看门狗定时器本质上是一个计数器，它会不断递减。如果计数器减到零，系统会认为程序运行异常，并触发复位操作。为了防止复位，程序需要定期“喂狗”（即重置计数器）。如果程序正常运行，计数器会被定期重置；如果程序卡住或进入死循环，计数器会减到零，触发复位。 3. Android Launcher定制：Launcher是平板开机后看到的第一个App。•用来显示和管理手机上所有已安装的app•显示显示和管理比较常用、重要的app•显示应用列表和快捷方式、小部件等等2、当按下Home键时，平板中显示的是你自己的桌面系统，而非系统自带的桌面系统3、本质上仍然是一个app，因此定制化Launcher的方式和开发一个普通app的流程基本一致。 | **结果展示**  拨码开关切换 UI 时，LED3（最简 UI）/LED4（正常 UI）同步点亮，数码显示当前温度值。  **问题展示**  CAN 接口与调试串口同时使用时出现中断优先级冲突，导致数据传输异常。  **困难展示**  硬件组与软件组对 “变姿控制信号协议” 理解不一致，引发多次代码返工。  **反省展示**  需补充学习嵌入式系统多线程编程与功耗管理专题，提升代码优化能力。  **合作共生展示**  优裁员主导代码评审，需求员根据评审意见调整 UI 交互逻辑，形成 “设计 - 开发 - 优化” 闭环。 |
| **发问** | **深度课设需求任务1·熟悉深度课设** | **展示** |
| **操作发问**  车联组队时，如何通过 LED 灯组（bit1-bit0）直观显示组队状态（单体 / 入队 / 头车 / 群驾）？  **编程发问**  看门狗机制（WDT）与 UI 界面的联动逻辑如何编码实现（如 ANR 检测时触发界面回滚）？  **任务发问**  如何设计轻量级数据持久化方案，确保看门狗触发前保存行驶里程、电池状态等关键数据？  **递进发问**  功耗优化从 “休眠待机” 到 “动态能耗互助”，需经历哪些迭代阶段？  **关联发问**  结伴运行逻辑中，如何通过线程同步确保多车状态实时同步？ | **向上**  **对齐**  **横向**  **对齐**  **深浅课设流畅对接**  **熟练运用五层结构，构造初步嵌入式系统**    **本组纸UI:**  48f8a5089d3f41fdbd5530c7a316d6e | **结果展示**  GitLab 记录完整代码迭代历史，关键模块（如组队逻辑）代码注释覆盖率达 90%。  **问题展示**  休眠模式下电流仍达 0.8A，远超设计目标（≤0.5A），经查为传感器未完全断电。  **困难展示**  实验箱接口数量有限，需频繁插拔模块，影响调试效率。  **反省展示**  建立每日站会制度，实时同步硬件状态与软件进度，减少信息差。  **合作共生展示**  向 “体联” 小组学习传感器低功耗设计经验，优化车联系统休眠模式电流。 |
| **发问** | **深度课设需求任务2·大版本建议要求** | **展示** |
|  | 二、授设同步递进·版本要求 **向上**  **对齐**  **横向**  **对齐**  **深浅课设流畅对接**  **熟练运用五层结构，构造初步嵌入式系统** |  |
| **操作发问**  变姿模式切换时（如平稳台→翘臀态），如何通过继电器接口控制硬件执行机构？  **编程发问**  在多车能量共享算法中，如何通过锁机制避免并发写操作导致的电量数据不一致？  **任务发问**  如何将 “除法法则” 应用于代码结构设计，实现硬件驱动与业务逻辑的彻底解耦？  **递进发问**  UI 设计从 “单屏显示” 升级为 “双屏交互”（触摸屏 + 16×16 点阵），如何优化数据同步机制  **关联发问**  浅度课设的双 UI 点名功能如何迁移至车联场景的温控与组队需求？ | **1.【浅度课设继承v1.X，权重25%】**   1. 双UI点名   中级UI：  输入设备：拨码开关用于选择操作模式，数字键盘用于输入具体信息。  显示设备：数码管显示当前操作状态，16×16 点阵显示组员信息。  提示设备：蜂鸣器用于提示操作完成或错误，LED 灯用于显示系统状态。  高级UI：  输入设备：触摸屏支持手势操作和手写输入。  显示设备：安卓大屏显示高清界面，支持动画效果和语音播报。  提示设备：立体声扬声器用于语音提示，震动马达用于触觉反馈。  多阶段系统  ⅰ.启动阶段：  硬件自检，加载系统配置，初始化 UI 界面，显示启动动画。  ⅱ.点名阶段（小组摘要&组员点名）  展示小组信息，组员依次点名，实时更新状态。  ⅲ.深度课设阶段  进入主题任务开发，支持功能扩展与优化。  浅度课设·双UI解耦问题递进（示范见附件一）  任务目标：实现双 UI 独立运行，可无缝切换，数据实时同步。  任务要求：  中级 UI：  实现拨码开关控制点名，数码管显示点名结果，点阵展示动画效果。  支持多级菜单，通过拨码开关切换不同功能模块。  高级 UI：  实现触摸屏点名，界面友好，支持语音提示与震动反馈。  支持手势操作和多点触控。  切换机制：  两种 UI 可自由切换，切换时数据无缝同步。  支持热插拔，允许在运行时更换 UI 设备。  数据同步：  实时同步点名数据，确保两种 UI 显示一致。  支持数据备份与恢复，防止数据丢失。  **2.【主题设计构筑v2.X，权重50%】**  **<主要任务：I. I/O主题设置 II.纸UI&屏UI设计· III.样例完善 IV.本组化 V.特色化>**   1. 主题化描述   1. I/O 主题设置  任务目标：根据主题需求，合理配置 I/O 资源。  任务要求：  输入设备：  数字键盘：用于输入具体信息。  拨码开关：用于选择操作模式。  触摸屏：支持手势操作和手写输入。  输出设备：  数码管：显示当前操作状态。  16×16 点阵：显示组员信息。  LED 灯：显示系统状态。  蜂鸣器：用于提示操作完成或错误。  安卓大屏：显示高清界面，支持动画效果和语音播报。  接口设计：  设计 I/O 接口电路，确保设备稳定运行。  支持热插拔与故障检测，确保系统可靠性。  2. 纸 UI&屏 UI 设计  任务目标：设计纸 UI 和屏 UI，提升用户体验。  任务要求：  纸 UI：  设计简洁明了的界面，突出关键信息。  支持手写输入与涂鸦功能，增加用户互动性。  提供详细的使用说明，方便用户快速上手。  屏 UI：  设计美观大方的界面，操作便捷。  支持多点触控与手势操作，提升交互体验。  提供语音提示与震动反馈，增强用户感知。  3. 样例完善  任务目标：完善样例代码，提升代码质量。  任务要求：  代码优化：  分析样例代码，找出问题与不足，进行优化与改进。  优化算法，提升代码运行效率。  修复潜在的漏洞，确保代码稳定性。  注释添加：  添加详细注释，提升代码可读性。  支持多语言开发与调试，方便团队协作。  提供代码示例与说明文档，方便其他成员学习。  4. 本组化  任务目标：结合小组特点，进行个性化设计。  任务要求：  分析小组优势：  分析小组成员的技术特长。  确定小组在项目中的优势方向。  个性化设计：  实现个性化功能，如智能调度、节能管理。  优化系统界面，突出小组特色。  提供定制化配置选项，满足不同用户需求。  5. 特色化  任务目标：打造小组特色，提升系统影响力。  任务要求：  特色功能设计：  结合主题需求与小组特点，设计特色功能。  实现车队能量共享、自动路径规划等创新功能。  功能实现：  完成特色功能的开发与测试，确保功能稳定运行。  提供详细的使用说明与文档，方便用户使用。  支持多场景应用与推广，提升系统价值。 | **结果展示**  继电器接口成功驱动模拟变姿机构，16×16 点阵实时显示车辆姿态图标  **问题展示**  ****通信延迟****：CAN 总线模拟数据传输延迟达 200ms，导致多车速度同步滞后。  **困难展示**  对嵌入式系统中实时操作系统（RTOS）任务调度机制理解不足，导致多车协同逻辑开发受阻。  **反省展示**  ****技术规划不足****：应在课设初期预留 RTOS 学习时间，避免后期功能实现瓶颈。  **合作共生展示**  向 “体联” 小组借用重力传感器数据校准算法，提升车联姿态检测精度。 |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **不同主题乘放大效能对比**  **完成三法则嵌入分析与目标指向递进** | **深度课设需求任务2·大版本建议要求** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | **车联Ⅰ：Eo-samrt（参照往届典型主题课设）**  **注意自动驾驶和物联电车的行业调研！**  通过车联网控制电动车的自动驾驶与道路规划，构建EO-samrt（该主题小组，需要多次观看相关视频和文档）未来电动物联车组队电控驾驶&规划的交通嵌入式系统。  【物联指向】降低出行代价难度·提高低碳出行体验·降低城镇交通负荷  变姿·结伴·共享 【车环境温控】 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **不同主题乘放大效能对比**  **完成三法则嵌入分析与目标指向递进** | **深度课设需求任务2·大版本建议要求** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | * 变姿：四轮传统正轮态/360旋转轮态+平稳台/翘臀态； * 结伴：单体运行+结伴运行+动态入队/离队； * 群驾：人工头车·跟随+准人工·跟随+激励策略（头车激励+能量互助）   该主题课设中建议，构造最简结伴规划算法。 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **不同主题加回滚合理自然比对**  **完成三法则嵌入分析与目标指向递进** | **深度课设需求任务2·大版本建议要求** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | 深度课设·主题化问题递进（示范见附件一）   1. 本组化描述   本组在广泛搜集资料、对象分析、群组讨论的基础上，完成任务规划，设计本组主题的具体实现方案  深度课设·本组化问题递进【权重5%】   1. 特色化描述   以更好满足主题需求为目标，设计实现具有本组智慧与特色的UI、策略方案与功能扩展  深度课设·特色化问题递进【权重5%】  **3.【可测可靠加入v3.X，权重10%】**  **<主要任务：I.主题回滚设计 II.WDT模板填充· III.给定异常与缺陷注入 >**  主题回滚逻辑  1. 主题回滚设计  任务目标：实现系统回滚功能，提升系统可靠性。  任务要求：  回滚触发条件：  分析系统运行状态，确定回滚触发条件。  设计回滚机制，支持数据备份与恢复。  回滚机制设计：  实现系统状态的实时监控，检测到异常时自动触发回滚。  支持多级回滚，根据错误严重程度选择回滚级别。  提供回滚日志，记录回滚操作与原因。  2. WDT 模板填充  任务目标：引入看门狗机制，提升系统抗干扰能力。  任务要求：  看门狗参数设置：  分析系统运行环境，确定看门狗参数。  设计看门狗机制，支持系统自恢复。  看门狗功能实现：  实现看门狗功能，定期检测系统状态。  检测到系统异常时，自动重启系统，确保系统稳定运行。  提供看门狗日志，记录异常检测与恢复操作。  3. 给定异常与缺陷注入  任务目标：通过异常与缺陷注入，提升系统测试效果。  任务要求：  异常与缺陷注入方案：  分析系统功能，设计异常与缺陷注入方案。  支持多种异常类型，模拟真实运行环境中的问题。  注入功能实现：  实现异常与缺陷注入功能，支持系统测试与优化。  提供详细的测试报告，记录异常注入与系统响应情况。  根据测试结果，优化系统代码，提升系统质量。 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **不同主题加回滚合理自然比对**  **完成三法则嵌入分析与目标指向递进** | **深度课设需求任务2·大版本建议要求** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** |  | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **不同主题加回滚合理自然比对**  **完成三法则嵌入分析与目标指向递进** | **深度课设需求任务2·大版本建议要求** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | 1. 异常边界可测修正   对异常、出错等情况能够及时判断并加以适当处理。   1. 看门狗修正   引入看门狗（Watchdog）机制，主动防御外界干扰，使系统拥有崩溃自恢复的功能，从而能够长时间不间断地运行，提高系统的抗干扰能力和可靠性。   1. 缺陷注入 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **小版本递进中的精进闭环工具对齐**  **运用三栏OKR与裁剪先进工具能力** | **深度课设需求任务2·大版本建议要求** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | 通过适当的测试源，有针对性地验证系统功能是否完善可靠，对系统开发设计进行审查，检验是否满足规定的需求或弄清预期结果与实际结果之间的差别。  深度课设·可测&靠问题递进（示范见附件一）  **4.【可优加入v4.X，权重15%】**  **<主要任务：I. 硬件优裁**·纸UI  **II. 软件优裁·系统定制/代码注释优化 III. 功耗优化 >**   1. 休眠待机功耗修正   - 在系统中加入休眠待机功能，优化系统能耗。  - 实现条件：系统在无操作时自动进入低功耗模式。  - 目标：减少系统在待机状态下的能耗，延长电池寿命。   1. 本组安卓定制修正   - 针对本组项目特点，对安卓系统进行定制化修改。  - 实现条件：定制化修改需不影响系统稳定性和兼容性。  - 目标：提高系统运行效率，优化用户体验。   1. 内存资源&代码优化   - 对系统内存资源进行优化，减少内存占用。  - 对代码进行注释优化，提高代码可读性和可维护性。  - 实现条件：优化后的代码需通过测试，确保功能正确性。  - 目标：提升系统性能，降低维护成本。  深度课设·可优问题递进（示范见附件一）  **5.【可联通加入v5.X】（选做）**   1. 蓝牙签字模拟联通   - 实现蓝牙设备之间的模拟联通，进行数据交换和签字确认。  - 实现条件：模拟联通需保证数据传输的安全性和准确性。  - 目标：通过蓝牙技术实现设备间的安全数据交换。   1. 数据迁移   - 实现系统数据的迁移功能，确保数据的完整性和可用性。  - 实现条件：数据迁移过程中需保证数据安全，防止数据丢失。  - 目标：方便用户在不同设备间迁移数据，提高数据的可移植性。  深度课设·可联问题递进（示范见附件一）  具体实施步骤：  1. 休眠待机功耗修正：  - 分析系统当前的功耗情况，确定优化点。  - 设计并实现休眠模式，包括定时器和唤醒机制。  - 测试休眠模式下的系统功耗，调整参数以达到最佳效果。  2. 本组安卓定制修正：  - 分析本组项目需求，确定需要定制的功能。  - 修改安卓系统源码，实现定制功能。  - 测试定制系统的功能和稳定性，确保满足项目需求。  3. 内存资源&代码优化：  - 分析系统内存使用情况，找出内存泄漏和冗余使用。  - 优化内存分配和回收机制，减少内存占用。  - 添加和优化代码注释，提高代码的可读性和可维护性。  4. 蓝牙签字模拟联通：  - 研究蓝牙通信协议，设计数据交换和签字确认流程。  - 实现蓝牙通信功能，包括设备配对、数据传输和安全认证。  - 测试蓝牙通信的稳定性和安全性，确保数据传输的可靠性。  5. 数据迁移：  - 设计数据迁移方案，包括数据格式、迁移工具和迁移流程。  - 实现数据迁移功能，包括数据备份、传输和恢复。  - 测试数据迁移的完整性和可用性，确保数据安全迁移。 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **深浅课设流畅对接**  **完成三法则嵌入分析与目标指向递进** | **深度课设需求任务3·系统量化指标对比测量** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | **三、量化指标·边测设置**  1、速度类指标（时域指标）  程序运行速度，UI响应速度（从操作触发到界面反馈的时间延迟）  2、功耗类指标  （最简/正常，不同模式、阶段）待机电流0.5A、最大电流1A、平均电流0.75A  3、内存资源类指标（选做）  内存占用（运行时动态内存使用情况）、IO资源占用（如串口、SPI 接口的使用频率与负载）  数据反思  **向上**  **对齐**  **横向**  **对齐**  **小版本递进中的精进闭环工具对齐**  **以计算定制培养诠释践行.南开公能校训** | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问** | **深度课设需求任务4·0操UI** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | **四、软硬结合三算·设计痕迹**  普算落地·边算设计·云算想象  任务目标：实现普算、边算、云算的有机结合，提升系统性能。  任务要求：  应用场景分析：  分析系统功能需求，确定普算、边算、云算的应用场景。  支持多平台部署与兼容，确保系统在不同设备上运行稳定。  协同机制设计：  设计普算、边算、云算的协同机制，支持数据交互与共享。  实现数据加密与压缩，确保数据传输安全。  提供多用户访问与共享功能，提升系统可用性。    1、0操设定与IO模拟  0操设定：  语境定义：  纸UI上面定义的文字内涵，用户0知识理解。  焦点选择：  屏幕画面上供用户可选择点不超过3个。  逻辑深度：  用户操作逻辑流程深度不超过2级。  IO模拟：  输入模拟：通过拨码开关、触摸屏等模拟用户输入。  输出模拟：通过数码管、点阵、LED 灯等显示系统状态。  2、云端数据消创（选做） | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **小版本递进中的精进闭环工具对齐**  **理解宏观嵌入式思维的消创数据生态** | **深度课设需求任务4·0操UI** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | 1. 车联Ⅰ：   节点数据消费设定  1. 数据类型  - 车辆状态数据：包括速度、位置、电量、故障代码等。  - 环境数据：如天气、交通状况、道路条件等。  - 用户行为数据：用户的操作习惯、偏好设置等。  2. 数据来源  - 车载传感器：直接从车辆的各类传感器收集实时数据。  - 外部API：通过与交通管理系统、天气服务等第三方服务的接口获取数据。  - 用户输入：通过车辆的用户界面收集用户的输入和反馈。  3. 数据消费流程  - 数据采集：定期或实时从上述来源收集数据。  - 数据处理：对收集到的数据进行清洗、过滤和格式化处理。  - 数据分析：利用数据分析工具对处理后的数据进行分析，提取有用信息。  - 数据应用：将分析结果应用于车辆控制、路线规划、用户通知等。  4. 数据安全与隐私  - 确保所有数据传输过程加密，防止数据泄露。  - 对敏感数据进行匿名化处理，保护用户隐私。  节点数据创造设定  1. 数据类型  - 预测性维护数据：基于车辆运行数据预测潜在故障。  - 优化行驶路线：根据实时交通和环境数据优化行驶路线。  - 用户个性化推荐：根据用户行为数据提供个性化服务推荐。  2. 数据创造流程  - 数据生成：利用机器学习算法和模型生成预测性维护数据和优化行驶路线。  - 数据整合：将生成的数据与现有数据整合，形成完整的数据集。  - 数据验证：通过实际运行验证生成数据的准确性和有效性。  - 数据发布：将验证后的数据发布到系统中，供车辆和用户使用。  3. 数据创新应用  - 智能调度：根据车辆状态和用户需求智能调度车辆。  - 节能驾驶：通过优化行驶路线和驾驶建议帮助用户节省能源。  - 安全预警：利用预测性维护数据提前发现并处理潜在故障，提高行车安全。  4. 数据管理与维护  - 定期更新数据模型和算法，确保数据生成的准确性。  - 建立数据质量监控机制，及时发现并纠正数据错误。  - 持续收集用户反馈，不断优化数据创造流程。   1. 车联Ⅱ：   节点数据消费设定  节点数据创造设定   1. 纸联Ⅲ：   节点数据消费设定  节点数据创造设定   1. 体联Ⅳ：   节点数据消费设定  节点数据创造设定  3、云算赋能（选做）   1. 车联Ⅰ云算赋能：   任务目标：利用云计算提升系统功能，支持多场景应用与推广。  任务要求：  探索云计算在系统中的应用场景，如计算加速、资源共享等。  实现云算赋能功能，提升系统性能。       1. 车联Ⅱ云算赋能： 2. 纸联Ⅲ云算赋能： 3. 体联Ⅳ云算赋能： | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **小版本递进中的精进闭环工具对齐**  **运用三栏OKR与裁剪先进工具能力** | **深度课设需求任务5·双构切换** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | **五、两套UI·内外双构**  1、外中级UI：温控（通俗可以理解为最简外中级UI温控，内高级UI主题系统）  设计目标：通过实验箱物理接口实现基础控制，适应低功耗、强稳定性场景，确保在触屏故障或极简操作需求下系统仍可运行。  1.1 数码管：实时状态可视化  U19-U20（前两位）：温度显示  数据来源：通过实验箱 AD 采集电位器模拟环境温度（0-5V 电压对应 0-100℃），经 ADC 转换后取整数部分显示。  显示格式：两位十进制数，不足十位补 “0”（如温度 25℃显示 “25”，5℃显示 “05”）。  U21-U22（后两位）：时间显示  计时逻辑：系统启动后从 “00” 开始累加，每分钟递增 1，最大显示 “99” 后循环归零。  应用场景：用于测试系统连续运行时间，或作为点名响应超时计时（如超过 30 秒未响应则触发警告）。  1.2 16×16 LED 点阵：状态符号化提示  分区设计：  将点阵划分为 4 个 8×8 子区域（左上、右上、左下、右下），每个区域独立显示一个大写字母或符号。  显示规则：  单字母模式：显示车辆编号（如 “C01” 表示组长车，“M02” 表示成员车 2）、状态码（“ST”= 休眠，“RD”= 行驶中）。  组合模式：通过多个子区域拼接显示简单图标（如 “→” 表示前进方向，“⚡” 表示能量共享）。  技术实现：  使用实验箱提供的点阵驱动库，通过二进制数组定义字符点阵（如字母 “A” 的点阵数据为0x01, 0x03, 0x05, ...）。  动态刷新：每 500ms 更新一次显示内容，避免高频刷新导致硬件损耗。  1.3 8 位拨码开关：物理化指令输入  状态定义：  On（向左拨动）：对应二进制 “1”，表示功能激活或参数开启。  Off（向右拨动）：对应二进制 “0”，表示功能关闭或参数复位。  功能分配：  1-7 号开关：模拟车辆激活状态（每一位对应一辆车）。  例：开关 1 打开→车辆 1 激活（LED1 灯亮），开关 2 关闭→车辆 2 休眠（LED2 灯灭）。  8 号开关：UI 模式切换（核心功能）。  向左拨动→进入 “最简 UI”（LED3 灯亮，LED4 灯灭），此时仅支持硬件按键操作。  向右拨动→切换至 “正常 UI”（LED4 灯亮，LED3 灯灭），触屏功能激活。  1.4 蜂鸣器与扬声器：多模态反馈  蜂鸣器：高频警告提示  短鸣（100ms）：操作确认（如拨码开关切换成功、点名响应）。  长鸣（500ms）：异常警告（如电量低于 20%、按键触发看门狗）。  立体声扬声器：语音交互  功能：配合点阵显示提供语音提示（如 “车辆 1 已激活”“休眠模式启动”）。  技术实现：通过 Android TextToSpeech引擎将文本转换为语音，音量通过实验箱物理按键调节。  1.5 AD 采集电位器：模拟量输入  功能：手动调节模拟温度值，用于测试系统温控逻辑。  操作方式：  顺时针旋转→电压升高→显示温度值增大（如从 20℃升至 60℃）。  逆时针旋转→电压降低→显示温度值减小（如从 60℃降至 20℃）。  联动逻辑：  温度≥80℃时，蜂鸣器长鸣，LED1-LED2 显示 “10”（故障阶段）。  温度≤10℃时，自动触发休眠提示（点阵显示 “SLP”，扬声器播报 “温度过低，即将休眠”）。  2、内高级UI：主题  内高级 UI（Normal）：主题区深度设计  设计目标：围绕 “车联 I・Eo-Smart” 主题，构建智能化、场景化的交互界面，通过分层布局实现功能解耦，提升操作效率与用户体验。  2.1主题区：0 操 UI 核心设计  2.1.1 语境定义  场景化术语：  界面文字均采用车联领域专业词汇，如 “头车”“能量共享”“变姿控制”“入队减速” 等，避免通用表述导致的歧义。  示例：  “组建车队” 按钮：点击后触发多车协同逻辑，而非简单 “创建群组”。  “变姿” 而非 “模式切换”：明确指向车辆行驶姿态的物理变化（如正轮态→旋转态）。  视觉隐喻：  用方向盘图标表示 “控制”，电池图标表示 “能量共享”，地图图标表示 “路径规划”，通过图形化符号降低认知成本。  2.1.2 焦点选择  功能入口控制：  主界面仅保留 3 个核心焦点按钮：  ① 组建 / 解散车队（核心功能，解决车联协同需求）  ② 车辆详情（查看实时数据，支持触控交互）  ③ 休眠设置（功耗优化，符合嵌入式系统特性）  动态焦点切换：  进入 “车队管理” 场景时，焦点自动切换至 “头车控制”“能量共享” 等子功能，通过底部标签栏（TabLayout）实现快速导航。  2.1.3 逻辑深度  操作层级限制：  所有功能操作流程不超过 2 级，采用 “主界面→子界面（弹窗 / 浮窗）” 结构。  示例：  组建车队：主界面点击 “组建车队”→弹出车辆选择浮窗（单选 / 多选）→确认后立即生效。  变姿控制：主界面长按 “变姿” 按钮→滑动选择姿态（正轮态 / 旋转态）→松手后实时生效。  异常处理扁平化：  错误提示直接显示在操作按钮下方（如 “车辆未激活，请先点名”），避免跳转新页面。  2.2 “×”、“＋” 分析映射初步实现  “×”（除法则：简化与裁剪）  功能精简：去除冗余操作（如传统车控中的 “灯光控制”），聚焦核心场景（组队、能量管理、路径规划）。  界面降噪：隐藏非必要元素（如系统级设置入口），仅在长按界面空白处时显示 “高级设置” 浮动按钮。  “＋”（加法则：扩展与增强）  功能叠加：在基础点名功能上增加 “多车协同启动”（V2.0）、“车队能量共享”（V3.0）等扩展功能。  数据聚合：在车辆详情页整合速度、电量、位置、姿态等多维度数据，通过卡片式布局（CardView）呈现。  2.3 点名区：动态状态反馈  标签下沉设计：  点名结果以半透明标签形式显示在屏幕底部，包含：  车辆编号：如 “C01”“M02”，颜色区分角色（蓝色 = 头车，绿色 = 成员车）。  响应状态：“已激活”“未响应”“超时”，搭配图标（√/×/⏳）快速识别。  操作按钮：直接在标签上滑动选择 “启动”“休眠”，无需返回主界面。  动态交互：  点名时标签从屏幕底部滑动弹出，3 秒后自动隐藏；长按标签可固定显示。  技术实现：使用 Android Snackbar组件实现可交互标签，通过CoordinatorLayout管理层级关系。  2.4 本组定制区：版本与团队标识  系统版本递进显示：  展示形式：界面右上角显示版本号（如 “V5.0”），点击后弹出版本日志浮窗，记录关键迭代：  V1.0→V2.0：从单一点名到多车协同（“×”→“＋” 的应用）。  V3.0→V5.0：加入看门狗、休眠、路径规划等可靠性与智能化功能（“可靠构造→可优转台”）。  团队标识：  界面底部显示 “NK2025 车联 I・第二组” 字样，点击后显示成员列表（巩岱松、郭子涵等）及分工，强化小组归属。  2.5 指标区：数据驱动决策  实时监控模块：  指标类型 显示内容 技术实现  速度类 头车速度、车队平均速度 TextView实时更新，单位 km/h  功耗类 休眠前后电流（如 “1A→0.5A”）、电量百分比 调用PowerManager接口获取数据  内存资源 内存占用率、CPU 使用率（选做） ActivityManager统计数据  自定义 风阻系数、轮胎磨损度（模拟值） 算法计算后显示（如变姿时动态变化）  可视化组件：  电量显示：环形进度条（CircularProgressBar），红色预警区域（低于 20%）。  速度趋势：迷你折线图（MPAndroidChart库），显示近 5 分钟速度波动。  2.6 互联区：设备与系统协同  车队互联设置：  入队 / 出队管理：输入车辆编号后点击 “加入车队”，头车通过蓝牙广播同步指令（选做，PPT 中未实现，计划 V6.0 扩展）。  能量共享配置：滑动调节共享比例（如头车向成员车传输 20% 电量），点击 “启动共享” 后实时生效。  硬件互联状态：  显示实验箱连接状态（如 “USB 调试已连接”“数码管正常工作”），异常时按钮变为红色并闪烁。  2.7 特色区：创新功能落地  多姿态控制：  操作入口：主界面 “变姿” 按钮，滑动选择姿态图标（正轮态、旋转态、翘臀态）。  联动效果：  界面显示对应动画（如旋转态车辆图标 360° 转动）。  硬件反馈：蜂鸣器模拟不同行驶音效，LED1-LED2 显示姿态编码（如旋转态 = 10）。  路径规划演示：  在地图区输入起点与终点（如 “八里台站→津南站”），点击 “规划路线” 后显示最短路径（基于 PPT 中MapPathfinder类的 Dijkstra 算法），并在实验箱点阵显示路径节点（如 “A→B→C”）。  3、自由切换  最简/正常UI自由切换  最简 / 正常 UI 可通过硬件拨码开关（8 号）或软件界面按钮自由切换，切换时自动同步状态并提供视觉 / 声音反馈：  触发方式：  硬件：拨动实验箱 8 号拨码开关（左→最简 UI，右→正常 UI）。  软件：在正常 UI 界面点击 “模式切换” 按钮（需长按 2 秒确认）。  状态同步：  切至最简 UI 时，自动读取硬件状态（如激活车辆、温度）并显示于数码管 / 点阵，关闭触屏等非必要功能以降低功耗。  切回正常 UI 时，恢复休眠时间、车队配置等软件设置，同步硬件状态至界面（如车辆激活状态实时更新）。  反馈机制：  视觉：实验箱 LED3（最简）/LED4（正常）灯亮，界面顶部显示模式标签（如 “当前模式：最简”）。  声音：切换时蜂鸣器短鸣，扬声器语音播报 “已切换至正常模式”。  冲突处理：硬件操作优先级高于软件，避免同时操作导致逻辑冲突（如拨码开关切换时强制覆盖软件设置）。  4、主题双构 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **小版本递进中的精进闭环工具对齐**  **运用三栏OKR与裁剪先进工具能力** | **深度课设需求任务5·双构切换** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | “主题双构” 指围绕 “车联 I・Eo-Smart” 核心主题，构建 \*\* 外中级 UI（最简模式）与内高级 UI（正常模式）\*\* 双架构，分别适配不同场景需求，实现 “硬件基础控制” 与 “软件智能管理” 的深度协同。  4.1 外中级 UI（最简模式）—— 物理交互为核心  定位：面向硬件调试、紧急控制场景，通过实验箱物理接口实现基础功能。  核心特性：  硬件直控：通过 8 位拨码开关选择车辆、数码管输入指令、点阵显示状态（如 “C01” 表示组长车激活）。  低耗稳定：关闭触屏和动画功能，仅保留必要硬件驱动，休眠时电流低至 0.5A。  极简逻辑：操作流程不超过 2 步（如拨码开关选车→按键点名），适合无软件环境下的快速操作。  4.2 内高级 UI（正常模式）—— 智能场景为核心  定位：面向日常使用场景，基于 Android 系统实现车联功能的智能化与可视化。  核心特性：  主题化交互：界面元素紧扣车联场景（如 “组建车队”“能量共享” 按钮），通过地图视图、车辆动画（如变姿旋转）增强沉浸感。  数据驱动：实时显示速度、电量、路径等数据，支持图表化展示（如电量消耗折线图），并通过 Dijkstra 算法实现最短路径规划。  多车协同：支持头车管理、车队能量共享、动态入队减速等功能，通过软件广播机制同步多车状态。  4.3 双构联动逻辑  模式切换：  通过 8 号拨码开关或软件按钮切换，切换时自动保存 / 读取状态（如休眠时间、车队配置）。  示例：从正常 UI 切至最简 UI 时，硬件数码管自动显示当前温度，软件保存的车队信息在切回后自动恢复。  功能互补：  最简 UI 作为 “保底控制”，确保软件故障时仍可通过硬件操作车辆；正常 UI 作为 “主力界面”，提供复杂场景下的智能化体验。  场景案例：  车辆维修时用最简 UI 单点控制硬件，排查故障；  日常出行时用正常 UI 规划车队路线，开启能量共享降低功耗。  4.4 设计价值  可靠性：双架构避免单一交互失效风险，符合嵌入式系统 “安全保底” 需求。  扩展性：最简 UI 提供硬件操作底层接口，正常 UI 通过软件迭代扩展功能（如从 V1.0 点名到 V5.0 自动驾驶预研）。  用户体验：物理按键与触屏操作并存，覆盖专业人员与普通用户的差异化需求，体现 “0 操 UI”（用户零学习成本）设计理念。 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **深浅课设流畅对接**  **运用三栏OKR和裁剪先进工具能力** | **深度课设需求任务5·双构切换** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | 深度课设通识纸UI布局建议：    **本组纸UI:**  **48f8a5089d3f41fdbd5530c7a316d6e** | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问** | **深度课设需求任务6·本组特色** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | **六、本组化定制痕迹**  **1、点名本组**  “点名本组” 指项目中体现的团队专属标识、成员联动机制或个性化点名功能。  1.1 团队身份可视化  Logo 与名称：  项目名称 “Eo-Smart” 结合 “电动物联”（Electro-Oriented Smart）概念，体现小组对车联主题的聚焦。  汇报材料中多次出现 “NK2025 嵌入式 - 第二组” 字样，并在 UI 界面（如 Launcher、登录页）嵌入小组名称，强化团队归属。  成员信息绑定：  登录界面显示 当前用户角色（如 “组长：巩岱松”），操作记录与成员账号关联，例如看门狗触发次数、休眠时间设置等数据均绑定至具体用户，实现操作可追溯。  1.2 群组点名功能定制  硬件与 UI 联动：  通过 蜂鸣器发声 配合 数码管 / 点阵显示 被点名车辆信息（如车号、成员姓名），例如 V1.0 版本实现 “蜂鸣器群组点名，数码管显示被点名者信息”，点名逻辑中优先触发组长车辆，体现团队层级。  点名界面采用 小组设计背景（如南阁大学元素、车联主题图标），区别于通用模板，增强视觉辨识度。  多车协同点名：  V2.0 版本支持 “多辆车共同点名启动”，通过 Android 广播机制同步点名指令，确保小组内车辆状态一致，例如组长触发点名后，成员车依次响应并显示 “已激活” 状态，体现团队协作特性。  **2、安卓本组**  “安卓本组” 指基于 Android 系统的个性化开发，包括系统适配、UI 定制与功能扩展。  2.1 Android 系统深度适配  硬件驱动开发：  针对实验箱硬件（如 S5P6818 处理器、触摸屏）开发定制驱动，实现 按键事件监听（如过快按键触发看门狗）、电源管理接口（休眠功能调用PowerManager）等底层功能，确保 Android 系统与硬件协同运行。  通过 JNI（Java Native Interface） 调用 C/C++ 编写的硬件控制代码（如蜂鸣器驱动），提升执行效率。  系统服务定制：  创建 后台服务（如CarManagerService）管理车辆状态，监听电量变化、位置更新等事件，并通过BroadcastReceiver通知 UI 界面实时刷新，例如电量低于阈值时触发全局广播，各车辆界面同步显示警告。  2.2 UI 界面个性化设计  Launcher 定制：  设计 小组专属 Launcher，背景采用南阁大学校园元素（如八里台站、津南站地标），图标样式统一为科技感风格（如齿轮、电路板图标），区别于 Android 原生 Launcher。  Launcher 布局分区明确：顶部显示小组名称与成员列表，中部为功能入口（组建车队、能量共享等），底部显示车辆状态快捷栏（电量、速度、位置）。  交互逻辑优化：  手势操作：在车辆详情页支持左右滑动切换成员车信息，符合 Android 用户操作习惯；长按功能按钮（如 “能量共享”）弹出二次确认弹窗，防止误触。  动态反馈：变姿操作时（如从 “正轮态” 切换为 “旋转态”），UI 界面显示车辆动画旋转，同时通过 Android 的ValueAnimator实现状态切换过渡效果，提升操作流畅度。  2.3 功能模块 Android 化实现  基于 Activity 的界面管理：  采用 单 Activity+Fragment 架构，实现界面解耦。例如，点名界面（RollCallFragment）、车队管理界面（FleetManagerFragment）等作为独立 Fragment，通过 Activity 管理生命周期，减少内存占用。  数据持久化：  使用 SharedPreferences 存储用户设置（如休眠时间、默认头车），确保重启后配置保留；通过 Room 数据库 记录操作日志（如看门狗触发记录、入队历史），支持按成员账号查询统计。  **3、主题本组**  “主题本组” 指围绕 “车联 I” 主题的专属设计，包括功能定位、视觉风格与技术选型。  3.1 功能聚焦车联场景  核心功能闭环：  针对车队协同场景开发 “组建 / 解散车队”“能量共享”“头车激励” 等功能，支持多车实时状态同步（如头车速度变更时，成员车自动调整速度保持车距），解决传统单车控制无法满足的团队出行需求。  实现 “入队 / 出队流程优化”，入队前车队主动减速，确保新车辆安全加入；出队时触发路径重规划，避免影响其他成员车行驶，体现车联场景下的协同控制逻辑。  差异化功能设计：  多姿态控制：车辆支持 “平稳态”（低功耗）、“旋转态”（灵活转向）等模式，对应不同的风阻系数和电量消耗，例如在限速区自动切换为平稳态，在弯道较多路段切换为旋转态，提升复杂路况适应性。  地图与路径规划：集成 Dijkstra 算法 实现最短路径计算，结合车联数据（如各车实时位置）动态调整路线，例如头车根据成员车电量分布选择最近充电点作为途经点，强化车联主题的智能化特性。  3.2 视觉风格统一车联元素  图标与配色：  采用 蓝色与灰色为主色调（象征科技与稳定），功能图标（如电池、方向盘、车队图标）均围绕车联主题设计，避免使用通用图标导致视觉混淆。  在车辆详情弹窗中，使用 仪表盘式电量显示、地图定位标记 等可视化元素，增强用户对车联状态的直观感知。  主题化交互反馈：  组建车队成功时，UI 界面弹出 “车队激活” 动画（多车图标汇聚成队列），并伴随蜂鸣器短鸣；能量共享开启时，界面显示电量流动特效，强化车联协同的视觉认知。  4、纸UI  “纸 UI” 指项目中纸质文档或汇报材料的可视化设计，体现团队风格与内容结构化。  48f8a5089d3f41fdbd5530c7a316d6e  4.1 汇报材料结构化设计  PPT 框架逻辑：  采用 “组员介绍→课设规划→成果展示→工具心得→关联认知” 的线性流程，符合技术汇报的逻辑顺序。每个章节设置明确标题（如 “PART 03 成果展示与设计提升”），通过目录页和页码（如 “01”“02”）引导听众理解内容层次。  在 “主题课设规划表” 章节使用 UML 模块图、流程分解图 等可视化工具，例如用流程图展示系统启动到车辆到达的完整流程，用表格对比 V1.0 至 V5.0 的功能迭代，提升信息传达效率。  视觉一致性：  PPT 背景统一使用 南阁大学 logo 与主题色（深蓝色），标题字体加粗并搭配车联相关图标（如小车、电路板）；正文采用简洁雅黑字体，避免复杂排版干扰内容聚焦。  4.2 文档细节定制  代码注释与版本记录：  在 Java 代码中添加 小组标识注释，例如 “// Modi Jin, 4 天前 | 1 author (Modi Jin)”，明确代码贡献者；在 GitLab 提交记录中注明 “[第二组] 优化休眠逻辑” 等信息，便于追溯小组专属开发痕迹。  实验报告特色：  在 “精进牵引” 章节插入 三栏 OKR 表格，左侧为目标（O），右侧分三列展示关键结果（KR）、负责人与进度，表格设计采用小组配色（蓝底白字），区别于通用报告模板。  在 “成果展示” 部分粘贴 纸质原型草图（如早期 UI 布局手绘稿），标注小组讨论时的修改意见（如 “此处需增加头车标识”），体现从概念到实现的迭代过程。  4.3 汇报交互设计  动态演示配合：  在 PPT 中嵌入 操作录屏 GIF（如看门狗触发时的系统回退动画）、代码片段截图（如休眠功能初始化代码），通过纸质材料与电子演示结合，增强汇报的立体感。  数据可视化：  用 折线图 展示休眠前后电流变化（从 1A 降至 0.5A）、柱状图 对比不同姿态下的电量消耗，图表标题标注 “第二组实验数据”，强化小组贡献标识。  **向上**  **对齐**  **横向**  **对齐**  **结伴小组示范对齐**  **以计算定制培养诠释践行.南开公能校训** | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问** | **深度课设需求任务7·主题底座与主题特色** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | **七、基础底座与主题特色痕迹**  车联I：点联降荷：变姿结伴·候鸟群慧  车联II：点联降本：判收降工·群资绿碳  纸联III：版联增场：异步共振·群组深学  体联IV：人联增康：健康节律·人人上医  1、横座竖题融合  任务目标：将基础底座与主题特色相结合，打造一体化系统。  任务要求：  1. 分析基础底座的功能与特性，确定其与主题的契合点。  - 基础底座功能：包括硬件接口、操作系统、通信协议等。  - 主题契合点：分析车联Ⅰ项目的具体需求，确定基础底座如何支持这些需求。  2. 在主题设计中融入基础底座的功能，实现功能互补。  - 硬件接口：设计兼容基础底座的硬件接口，确保设备的兼容性和扩展性。  - 操作系统：利用基础底座提供的操作系统特性，优化系统性能和稳定性。  - 通信协议：集成基础底座的通信协议，实现设备间的高效通信。  2、创新点指向要求  策略创新：提出新的系统运行策略，提升系统效率。  - 例如，引入智能调度算法，优化车辆的运行路径和时间。  UI 创新：设计新颖的用户界面，提升用户体验。  - 开发直观的图形界面，简化用户操作流程，提高用户满意度。  功能扩展：增加新的功能模块，丰富系统功能。  - 例如，增加远程监控和诊断功能，提高车辆的可管理性。  需求完善：根据用户反馈，完善系统需求。  - 定期收集用户反馈，分析用户需求的变化，及时调整系统功能。  3、对象与系统干扰分析  任务目标：分析对象与系统之间的干扰，提升系统稳定性。  任务要求：  1. 任务分析 ：  确定系统中的关键对象及其特性。  - 识别系统中的关键硬件组件、软件模块和用户交互点。  2. 设计规划：  分析对象与系统之间的相互作用，找出潜在干扰。  - 分析这些对象如何影响系统的性能和稳定性，识别可能的干扰源。  3. 初步实现  提出解决方案，减少干扰，提升系统稳定性。  - 设计抗干扰措施，如屏蔽、滤波、错误检测和纠正机制。 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **小版本递进中的精进闭环工具对齐**  **运用三栏OKR与裁剪先进工具能力** | **深度课设需求任务8·其他综合** | **展示** |
| **操作发问**  在多种运行模式共存的嵌入式控制系统中，模式切换过程中是否可能存在冲突或转场代价？系统是否具备有效的状态缓冲机制与切换确认逻辑，以保障切换过程的稳定性与安全性？  **编程发问**  某些模式是否需要禁止休眠（如远程协助、医疗呼叫）？如何在设计中协调模式与功耗控制的优先级？  **任务发问**  在休眠机制中新增 “外部中断唤醒” 功能（如按键、传感器触发），是否会增加系统复杂度？如何平衡扩展与稳定性？  **递进发问**  是否构建了“功耗画像”？即哪些子系统在何种模式下的电量消耗预测？  **关联发问**  模式切换是否存在冲突或转场代价？是否有状态缓冲与确认机制？ | **八、可嵌入系统整体性痕迹（选做）**  1、系统构造代价合理性分析  系统构造代价合理性需从硬件选型、软件开发、人力成本、时间成本及功能实现的性价比等维度综合评估。  1.1硬件成本与性能适配  处理器选择：采用 S5P6818 处理器（ARM Cortex-A53 核心，ARMv8-A 架构），具备多核心处理能力和丰富接口（如 USB、UART、SPI 等），可满足车联控制板的实时数据处理、多任务调度需求。该芯片在嵌入式领域应用广泛，性价比高，适合中小型项目，硬件成本可控。  功能模块扩展：  输入设备：集成数字键盘、拨码开关、触摸屏，覆盖多样化操作场景（如手动按键控制、触屏交互），成本随接口复杂度增加但用户体验提升显著。  输出设备：实验箱屏幕、蜂鸣器、数码管、LED 灯等组合，实现状态可视化（如电量、速度）和声光反馈，硬件成本较低且功能明确。  成本优化：通过 硬件裁剪（如优裁员负责模块精简），去除非核心功能组件，避免资源浪费，确保硬件成本控制在合理范围。  1.2软件开发与技术选型  语言与工具链：使用 Java 语言开发，依托 Eclipse 调试环境，适配 Android 系统（实验箱可能基于 Android），开发效率较高且社区资源丰富，降低学习成本。  版本管理与协作：采用 GitLab V18.0.1 进行代码管理，通过分支策略（如功能分支、修复分支）实现并行开发，减少冲突成本。团队成员可实时同步进度，提升协作效率，间接降低时间成本。  功能实现成本：  基础功能（如电量显示、看门狗、休眠功能）开发难度较低，依托现有框架和库可快速实现。  创新功能（如车队能量共享、Dijkstra 算法路径规划）需额外投入算法设计和调试时间，但通过模块化开发（如将地图路径计算封装为独立类），可复用代码降低后续迭代成本。  1.3人力与时间成本分配  团队分工：  组长统筹目标与进度，确保资源合理分配，避免重复劳动。  需求员、优裁员、周志员各司其职，分别负责需求梳理、代码优化、问题跟踪，专业化分工提升效率。  展示员与组助辅助文档整理和团队协作，减少沟通成本。  时间规划：  分阶段实现功能（V1.0 至 V5.0），逐步递增复杂度，避免一次性开发压力过大。例如，先完成基础点名功能（V1.0），再扩展至车队管理（V3.0），符合 “从简单到复杂” 的开发逻辑，降低时间浪费。  通过 三栏 OKR 管理，明确各阶段关键结果（KR），如 “实现蜂鸣器群组点名”“优化休眠功能”，确保时间投入与目标匹配，避免资源错配。  1.4性价比与功能价值  基础功能闭环：实现短途出行控制、电量管理等核心需求，满足用户基本使用场景，功能实用性强，成本投入回报明确。  创新功能溢价：  多姿态控制（平稳态、旋转态）和 路径规划算法 提升车辆适应性和智能化水平，虽增加开发成本，但可差异化竞争，提升项目附加值。  休眠功能降低功耗（休眠前后电流从 1A 降至 0.5A），延长设备续航，增强用户体验，长期使用中可节省能源成本。  2、本体对象控制与智能提升  本体对象控制指对系统核心实体（如车辆、车队）的状态管理与操作响应；智能提升则通过算法、自动化机制等增强系统自主性与适应性。  2.1 本体对象控制机制  车辆状态模型：  属性定义：每辆车包含 电量、速度、位置、行驶姿态（正轮态、旋转态等）、运行状态（活跃 / 休眠） 等属性，通过类（如Car类）封装，实现状态统一管理。  操作接口：  基础控制：启动 / 停止、变姿（setPosture()）、充电（模拟电量更新）。  车队协同：入队 / 出队（触发车队减速逻辑）、能量共享（头车向成员车分配电量）。  车队管理逻辑：  头车核心作用：作为车队主控节点，负责路径规划、速度协调（如 “车队减速转弯”）和能量调度，通过updateCarInfo()方法实时同步头车状态至成员车。  状态一致性：通过 GitLab 版本控制 和 共享数据结构（如List<CarState>），确保多车状态在分布式场景下的一致性，避免 “幽灵车” 等异常。  2.2智能化提升实现路径  算法驱动优化：  Dijkstra 算法路径规划：在MapPathfinder类中实现最短路径计算，输入起点、终点和路线数据，输出最优路径（List<MapLocation>），提升车辆导航效率，减少无效行驶能耗。  看门狗机制：通过按键次数阈值（MAX\_KEY\_COUNT=5）和超时时间（KEY\_TIMEOUT=2000ms），防止误操作导致系统崩溃，自动回退至安全状态并记录触发次数，增强容错性。  自动化与自适应：  休眠功能：根据用户设置的休眠间隔（30 秒至 10 分钟），通过PowerManager实现系统低功耗模式，休眠期间关闭非必要组件（如屏幕背光），仅保留看门狗监控，功耗降低 50%（电流从 1A 降至 0.5A）。  电量阈值响应：当电量低于设定值时，车队自动减速并提示充电，避免因电量耗尽导致故障，实现 “感知 - 决策 - 执行” 的闭环。  用户体验智能化：  UI 交互优化：通过 个性化 Launcher 界面、图标动画（如车辆变姿动态显示）和弹窗提示（showPopupWindow()），降低用户学习成本，提升操作直觉性。  群组点名智能化：支持多车批量启动，通过蜂鸣器与数码管联动，实现 “点名 - 响应” 自动化，减少人工操作步骤。  2.3控制与智能的协同效应  数据驱动控制：车辆状态数据（如速度、电量）实时同步至 UI 和后台，为路径规划、能量共享等智能功能提供决策依据。例如，头车根据成员车电量分布动态调整能量共享策略，避免过度消耗某辆车的电量。  异常处理闭环：  入队时检测成员车位置一致性，若偏离则抛出异常并提示调整，确保车队协同安全。  休眠前保存车辆状态（savedCarStates列表），唤醒后恢复现场，避免数据丢失，提升系统可靠性。  **九、课程与课设关联痕迹**  （一）课程知识应用  任务目标：将课程知识应用于课设中，提升实践能力，实现理论与实践的紧密结合。  任务要求：  1. 回顾课程中学习的嵌入式系统知识。  - 复习课程中的关键概念，如微处理器架构、实时操作系统、嵌入式编程、中断处理、内存管理等。  - 理解嵌入式系统的资源限制和设计挑战，如处理能力、内存大小、能耗等。  2. 在课设中应用这些知识，解决实际问题。  - 将理论知识应用于项目开发，如使用C语言编写嵌入式程序，使用RTOS管理任务调度，实现多任务并发执行。  - 利用中断处理机制提高系统的响应速度和实时性。  - 通过内存管理优化，确保系统在资源受限的情况下高效运行。  3. 创新实践，实现“小马拉大车”的效果。  - 在有限的硬件资源下，通过软件优化和算法创新，实现系统的高性能和高可靠性。  - 采用模块化设计和分层架构，提高系统的可维护性和可扩展性。  4. 借鉴“都江堰结构”的智慧，设计系统架构。  - 学习都江堰的分水、导流、排沙等原理，设计嵌入式系统的资源分配、任务调度和错误处理机制。  - 通过精细化管理，实现系统的稳定运行和高效性能。  （二）课设对课程的反馈  任务目标：通过课设反馈课程中的不足，提升课程质量，确保课程内容与实际需求的匹配。  任务要求：  1. 分析课设过程中遇到的问题，找出课程知识的不足之处。  - 识别课程中未覆盖或未充分讲解的知识点，如新兴技术、行业标准、实际开发工具等。  - 分析这些问题对项目开发的影响，评估课程知识与实际需求之间的差距。  2. 提出改进建议，完善课程内容。  - 建议课程增加实践环节，如实验室操作、项目开发、案例分析等，以增强学生的实践能力。  - 建议课程更新教学内容，引入最新的嵌入式系统技术和行业发展趋势。  - 建议课程加强与企业的合作，提供更多的实习和就业机会，让学生了解行业需求和工作流程。  3. 建立课程与课设的持续改进机制。  - 定期收集学生和教师的反馈，评估课程和课设的效果。  - 根据反馈结果，及时调整课程内容和教学方法，确保课程的持续改进和优化。  - 通过课程与课设的紧密结合，提高学生的综合素质和就业竞争力。  **十、gitlab·issue与认知OKR**  群组自我驱动·结伴协同互助·同类竞合引控   1. issue·递进lable·board：问题矩阵  |  |  |  | | --- | --- | --- | | **模块** | **核心要求** | **操作要点** | | ****GitLab Issue 管理**** | 建立问题矩阵：按 “功能开发”“UI 优化”“异常处理” 等分类创建 Issue 标注递进标签（如v1.0 bug enhancement） 使用 Board 看板跟踪 Issue 状态（待办 / 进行中 / 已解决） | - 组长负责统筹 Issue 分配，组员定期更新进度 - 每个 Issue 需关联代码提交记录，形成闭环追溯 | | ****认知 OKR 设定**** | ****团队 OKR****： O1：熟练运用五层结构构造嵌入式系统 O2：完成三法则（除、乘、加）嵌入分析 ****个人 OKR****： 组长：统筹进度与技术决策 需求员：精准提炼需求并文档化 优裁员：代码优化与功耗优化 | - 遵循 SMART 原则（具体、可量化、有时限） - 每周同步 OKR 完成度，通过 GitLab Wiki 记录反思 | | ****协同机制**** | 同类竞合引控：组内分工明确，避免重复开发；与其他小组良性竞争，共享通用模块 结伴协同：通过 Issue 评论区讨论技术细节，定期召开线上会议同步进展 | - 利用 GitLab 的 Merge Request 机制进行代码 review - 对共性问题创建跨组 Issue，协作解决 |   **向上**  **对齐**  **横向**  **对齐**  **结伴小组示范对齐**  **全面打开放开公开自我生命系统，示范群组主动、合作与探索性学习主动，** | **结果展示**  完成 EO-Smart 电动物联车控制板设计，实现 UI 界面设计、电量显示、看门狗机制、休眠功能等。  **问题展示**  初期休眠功能与看门狗机制存在逻辑冲突，导致系统不稳定  **困难展示**  在处理车队状态更新（如能量共享、位置同步）时，多线程并发导致数据不一致，需反复调试锁机制。  **反省展示**  对实验箱硬件特性和 GitLab 高级功能（如 CI/CD 流水线）掌握不深入，导致开发后期才引入版本冲突解决策略  **合作共生展示**  通过三栏 OKR 明确目标（O）与关键结果（KR），如 “提升系统智能化” 对应 “实现车队能量共享”（KR1），确保任务可量化、可追踪。 |
| **发问** | **深度课设需求任务8·展示反省** | **展示** |
| **操作发问**  如何添加微交互提升用户感知？如不同电量动画图标效果，充电完成后的弹窗显示，点击提交后的加载状态显示等？  **编程发问**  设计 “按钮点击效果” 时，除了颜色变化，如何添加微交互（如轻微缩放动画、点击音效）提升用户感知？  **任务发问**  设计一个 “下拉选择框” 组件时，如何优化键盘用户的操作体验（如通过上下箭头切换选项、回车键确认）  **递进发问**  模拟 “充电进度 UI” 时，如何通过 CSS 动画实现电池图标内的电量填充效果？充电完成后如何触发提示弹窗（结合 JavaScript 事件监听）？  **关联发问**  模拟 “电量显示” 时，如何设计 UI 元素（如电池图标）的状态切换逻辑（满电 / 低电量 / 充电中）？ | **十一、展示分享与反省**  1、三法则分析授设关联    完成“÷”、“×”、“＋”三法则分析，对深度主题课设映射模板进行填充   | ****三法则**** | ****定义与目标**** | ****车联 Ⅰ 主题实践案例**** | ****与课程 / 课设关联**** | | --- | --- | --- | --- | | ****除法（简化）**** | 剥离非核心功能，聚焦关键场景，降低系统复杂度，提升运行效率。 | - ****功能简化****： ▶ 初期剥离蓝牙签字、云端数据消创等选做功能，优先实现基础车联组队（单体运行→结伴运行）、变姿（四轮正轮态→360 旋转轮态）、温控（环境温度监测） ▶ 简化 UI 交互逻辑：仅保留 3 个焦点选择（“组队”“变姿”“温度调节”），操作流程深度≤2 级（如：拨码开关切换 UI→触摸屏选择功能） - ****硬件裁剪****： ▶ 暂不启用 CAN 接口、4G 模块，优先使用调试串口、LED 灯、数码管实现基础状态反馈 | - 对应课程五层架构中的****应用层****与****驱动层****解耦，通过简化应用场景聚焦底层硬件控制（如 LED 灯位操作、拨码开关状态读取） - 符合浅度课设 “双 UI 解耦” 基础，通过 “除法” 快速验证核心逻辑（如车联组队状态同步） | | ****乘法（放大）**** | 强化核心优势，通过功能扩展或性能优化放大系统价值，聚焦用户体验或效率提升。 | - ****核心功能放大****： ▶ 放大 “结伴运行” 场景：实现动态入队 / 离队逻辑，引入头车激励策略（头车优先分配电量），通过 LED 灯组（bit1-bit0）显示组队状态（00：单体，01：入队，10：头车，11：群驾） ▶ 优化温控逻辑：AD 采集电位器模拟温度输入，数码管实时显示温度值，当温度超过阈值时通过蜂鸣器报警并触发 LED3（最简 UI 标识灯）闪烁 - ****体验优化****： ▶ 在高级 UI（触摸屏）中增加动画反馈：车辆变姿时显示 360 旋转动画，组队成功时播放语音提示 “入队成功” | - 对应课程 “主题化设计构筑” 要求，通过 “乘法” 扩展浅度课设的点名功能，迁移至车联场景的组队协同 - 结合《嵌入式系统设计》中 “事件驱动编程” 理论，通过中断机制实现温度报警与 UI 动画同步 | | ****加法（扩展）**** | 引入新功能、交互或技术，增强系统完整性与扩展性，实现场景延伸或多维度价值。 | - ****功能扩展****： ▶ 增加 “能耗互助” 机制：当某车辆电量＜20% 时，自动向头车请求能量共享，头车通过 CAN 接口模拟能量传输（LED 灯组显示电量转移过程） ▶ 扩展异常处理：引入看门狗（WDT）机制，当系统卡死时自动重启并通过数码管显示错误代码（如 “ER01” 表示 UI 切换超时） - ****交互扩展****： ▶ 双 UI 联动：中级 UI（拨码开关 + 数码管）显示实时数据，高级 UI（触摸屏）展示历史能耗曲线，通过 8 号拨码开关实现一键切换 | - 对应课程 “可测可靠加入” 与 “可优加入” 要求，通过 “加法” 集成浅度课设的 “多阶段系统”（启动→点名→深度课设），并扩展至车联场景的故障自恢复（看门狗）、数据可视化（触摸屏曲线） - 体现《计算机组成原理》中 “存储层次结构” 思想，通过看门狗触发前保存关键数据（如行驶里程、电量状态） | | **结果展示**  通过蜂鸣器群组点名、数码管 / 点阵显示信息，实现多车协同控制  **问题展示**  当前仅基于预设地图数据计算路径，缺乏实时路况动态更新能力。  **困难展示**  实验箱部分外设（如触摸屏）驱动兼容性差，需自行修改底层代码实现稳定通信  **反省展示**  对实验箱硬件特性和 GitLab 高级功能（如 CI/CD 流水线）掌握不深入，导致开发后期才引入版本冲突解决策略  **合作共生展示**  定期开展组内技术分享（如 Java 多线程、Git 操作），何叶（组助）整理《开发问题手册》，避免重复踩坑 |
| **发问**  **向上**  **对齐**  **横向**  **对齐**  **结伴小组示范对齐**  **全面打开放开公开自我生命系统，示范群组主动、合作与探索性学习主动，** | **深度课设需求任务8·展示反省** | **展示** |
| **操作发问**  如何根据车辆当前的目的地和距离限制，动态匹配可加入的车队？是否需要定义一个“匹配窗口”或“地理围栏”来约束入队条件？  **编程发问**  在一个异构车队中，如何实时协调车速与电量分布，确保整体运行效率最优，同时防止低电量车辆成为瓶颈？  **任务发问**  在不同路况或任务目标下，车辆应如何切换变姿（如纵队、并排、楔形等）以适应高速行驶、节能行驶或道路狭窄等不同需求？  **递进发问**  是否允许电量不足的车辆强制“跳入”车队？在电量与目的地匹配不足的情况下，系统应如何提示或限制车辆入队？  **关联发问**  如果车主希望加入一个“节能优先”的队伍或“速度优先”的队伍，系统应如何在当前车队中标注其特性供车**主决策？** | 2、组成果展示  Ⅰ.展示视频（不大于5M）、照片  将最终完成的代码在嵌入式教学实验箱Android平台上操作运行，并对整个运行过程录像、拍照。  操作演示系统启动、UI 切换、功能验证（如车联组队、车队行进、选择最短路径等等），清晰呈现硬件交互（如拨码开关、触摸屏操作）  Ⅱ.简单成果展示ppt（成果/主要问题/授课交叉关联反省）  以ppt的形式展示和总结小组的最终成果、遇到的主要问题、取得的进步、有待改进的地方以及如何改进等。  - ****PPT 汇报****： 系统架构与核心功能，突出技术亮点与创新点 遇到的主要问题及解决方案 课程知识的关联（如五层结构应用）  Ⅲ.课设报告  文档结构：需求分析→设计方案→实现细节→测试数据→总结反思 - 重点内容： ▶ 双 UI 设计逻辑（中级 UI 与高级 UI 的解耦与切换） ▶ 三法则应用案例（如简化功能、放大核心场景、扩展交互）  3、组人学习反省  gitlab、issue使用心得，角色贡献  ****1.GitLab 使用心得****：Issue 管理效率、代码版本控制经验 2.****角色贡献评估****：组长协调能力、需求员需求分析准确性、优裁员优化效果 3.****改进计划****：针对系统漏洞（如 UI 卡顿）与协作瓶颈（如沟通延迟）提出具体措施  **本组纸UI:**  **48f8a5089d3f41fdbd5530c7a316d6e** | **结果展示**  车辆支持 “正轮态”“旋转态” 等多姿态，适应不同路况（如风阻、障碍），动态调整速度与电量消耗  **问题展示**  车队变姿控制时，部分车辆状态同步存在延迟，尤其在复杂网络环境下。  **困难展示**  Dijkstra 算法在大规模地图场景下计算效率低，需探索更高效的路径规划算法（如 A \* 算法）。  **反省展示**  初期目标（如 “实现 AI 能控・级联”）设定过于理想化，未结合团队实际技术储备，后期被迫调整。  **合作共生展示**  计划将本次课设的多车协同算法应用于 “空间移动机器人” 项目（2026-2028 年目标），深化嵌入式系统与 AI 融合。 |
| **发问** | **附件参考材料** | **展示** |
|  | 附件一【问题递进闭环·样例建议模板】浅度课设·双UI解耦群组点名提升 |  |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | 1. 【当前代码注释描述】   混合UI群组点名；  主题回滚；  车队能量共享；  显示车辆位置信息；  车队电源，速度，行驶距离，电量过低车队减速；  键盘控制车队；  看门狗和launcher界面；  休眠；  路径搜索，最小化能耗；  二、【issue问题递进发动】递进发问·讨论细化·归纳聚焦  <递进问题issue1:发起> 如何定义双UI解耦内涵理解？ 发起人 ：助教示范/组长本组自发 lable标识？  <递进问题issue2:细分> 中级UI，独立拨码开关模拟群组点名表达？ 发起人 ：组长本组自发 lable标识？  <递进问题issue3:细分> 高级UI，独立触摸大屏模拟群组点名表达？ 发起人 ：组长本组自发 lable标识？  <递进问题issue4:融合> 如何融合切换两种独立UI点名？ 发起人 ：组长本组自发 lable标识？  <递进问题issue5:提升> 怎样融合梳理两种UI点名的提升浅度课设·多阶段群组点名任务？发起人 ：组长本组自发 lable标识？  <递进问题issue6:授课课设结合>在车联I主题的实验中，我们通过简化系统功能（除法）、放大核心优势（乘法）和扩展交互维度（加法）来提高系统的效率与适应性。 在嵌入式系统设计中，“除法（简化）”“乘法（放大）”和“加法（扩展）”这三种设计思维 作为一种抽象的系统设计范式，是否具有普适性？ 是否存在某些设计场景中三种范式难以共存或彼此冲突的情况？ 本项目中， 在车联系统的迭代优化过程中，应如何动态选择或调整这些范式的应用优先级？ 发起人 ：组长本组自发 lable  <递进问题issue7:细分>如何在能量共享中实现能源的均衡分配？  在车队能量共享中，如何设计分配策略以实现能源的均衡分配？是否可以引入不同优先级的分配机制，例如在充电时优先为车头分配更多的电量，而其他车辆分配较少的电量？这种策略如何影响车队整体的运行效率和续航能力？ 发起人 ：组长本组自发 lable  <递进问题issue8:细分>如何在能量共享中实现能源的均衡分配？  在车队能量共享中，如何设计分配策略以实现能源的均衡分配？是否可以引入不同优先级的分配机制，例如在充电时优先为车头分配更多的电量，而其他车辆分配较少的电量？这种策略如何影响车队整体的运行效率和续航能力？ 发起人 ：组长本组自发 lable  <递进问题issue9:细分>如何将风阻减少的模拟与车辆编队策略结合？  在实际行进中，编队可以通过调整队形（如从单列队形调整为三角队形）来减少风阻。如何在代码中模拟这种队形调整的过程，并结合车辆的速度、电量消耗等参数，动态优化队形以模拟真实的行进状态？ 发起人 ：组长本组自发 lable  <递进问题issue10:细分>如何增加车辆的位置点以显示行进轨迹？  当前代码中车辆的移动仅通过速度和距离进行模拟，能否引入位置点（如二维坐标）来记录每辆车的实时位置，并在界面上显示车辆的行进轨迹？此外，如何设计数据结构以高效存储和展示车辆的历史路径？  发起人 ：组长本组自发 lable  <递进问题issue11:提升>在嵌入式系统开发过程中，从用户交互体验优化的角度出发，能否基于系统资源与性能限制，实现多样化的用户界面（UI）设计？同时，如何在嵌入式设备有限的硬件资源条件下，有效集成诸如淡入淡出、动态旋转、平滑缩放等丰富的动画效果？这些动画效果的实现又将对系统的响应速度、功耗等性能指标产生怎样的影响，以及如何通过技术手段平衡用户体验与系统性能之间的关系？ 发起人 ：组长本组自发 lable  <递进问题issue12:提升>基于嵌入式系统构建的智能车队管理体系中，能否通过权限分配机制实现精细化管控？具体而言，一方面，当车队领航车辆执行停车指令时，如何利用嵌入式系统的通信与控制功能，实现权限级联触发，使其余车辆同步响应停车操作；另一方面，针对车队中不同车辆与用户角色，能否依据职责需求，为其精准划分对车辆运行参数（如车速调节、姿态控制等）的操作权限，进而构建高效、安全且有序的车队协同运行模式？  发起人 ：组长本组自发 lable  <递进问题issue13:细分>能否使用其他的数据结构设计车队，以及车队的姿态，车速等属性  发起人 ：组长本组自发 lable  <递进问题issue14:授课课设结合>结合五层架构中的驱动层，如何将硬件控制部分（蜂鸣器、LED灯等）的代码进一步封装，从而为硬件控制和数据管理等部分设计统一的接口？  发起人 ：组长本组自发 lable  <递进问题issue15:授课课设结合>在城市智能出行背景下，如何构建一个支持车辆动态编队的分层嵌入式控制系统，使其能够在“目的地相同且距离不超过设定阈值”的条件下，智能判断是否允许车辆入队，并根据车速与剩余电量自动调整跟车策略与车辆姿态？具体而言：  感知层应采集关键传感数据（如 GPS 位置、电量、电机状态、当前速度等），以支持精准判断与策略执行  决策层利用上述信息，结合队伍整体目标，判断新车是否满足入队条件，并规划其最优入队路径与时机  执行层控制转向、加速、减速等变姿行为，确保合流过程的安全性与流畅性  整个系统的任务调度机制应如何设计，以在嵌入式资源受限的前提下，平衡实时响应能力与能源消耗效率  发起人 ：组长本组自发 lable  <递进问题issue16:授课课设结合>在车联领域的 "除法则" 框架下，需实现全向互联设计与驾控算子功能。当面临实验箱接口有限等硬件裁剪约束，以及需要通过入队距离限制等软件算法优化时，怎样在确保功能完整性的同时，有效控制并降低系统复杂度？  发起人 ：组长本组自发 lable  <递进问题issue17:提升>车主有加入 “节能优先” 车队或 “速度优先” 车队的意愿时，系统该采取何种方式，在现有车队信息中清晰标注出不同车队的特性，从而为车主提供有效的决策参考？  发起人 ：组长本组自发 lable  <递进问题issue18:提升>针对车联 I 的 “老年短途出行” 目标，当集成摔倒检测、一键呼叫等紧急救援功能时，怎样让这些功能与现有的车联网络相互协作，构建起完整有效的应急响应闭环体系？  发起人 ：组长本组自发 lable  <递进问题issue19:细分>关于UI切换时的防抖，通过拨码开关切换 UI 状态时（PPT 第 17 页），怎么避免硬件抖动或误操作导致系统紊乱？需要加软件防抖或确认弹窗吗？  发起人 ：组长本组自发 lable  <递进问题issue20:细分>关于多车队的电量管理，针对多车入队的电量激励目标（PPT 第 6 页），如何设计动态电量共享算法，防止单辆车电量过低拖垮整个车队？是否要加电量预警和自动脱离机制？  发起人 ：组长本组自发 lable  <递进问题issue21:提升>关于多车入队时的并发冲突的处理，多车同时申请入队时，系统怎么处理 ID 重复或路线冲突？是否需要按距离、电量等优先级来仲裁？  发起人 ：组长本组自发 lable  <递进问题issue22:提升>关于优化UI界面，如何添加微交互提升用户感知？如不同电量动画图标效果，充电完成后的弹窗显示，点击提交后的加载状态显示等？  发起人 ：组长本组自发 lable  <递进问题issue23:融合>如何设计看门狗与 Launcher 的系统级联动逻辑？具体来说，若车联 I 的 Launcher 负责界面调度和任务管理，看门狗检测到 ANR 或死锁时，除重启应用外，如何通过 Launcher 实现状态回滚（如恢复最近一次保存的行驶路线）？  发起人 ：组长本组自发 lable  <递进问题issue24:提升>车联 I 场景中，哪些关键数据需要在看门狗触发前优先保存？如行驶里程、电池状态、用户设置等，如何设计轻量级数据持久化方案（如 SharedPreferences 或本地数据库），确保在系统重启后快速恢复？  发起人 ：组长本组自发 lable  <递进问题issue25:提升>若看门狗触发时正在写入数据，如何避免文件损坏？是否需要引入事务机制或缓存队列？  发起人 ：组长本组自发 lable  <递进问题issue26:授课课设结合>对于“看门狗”机制得除乘加法则发问：  “除法”（简化核心功能）：是否需剥离看门狗中非必要的监控项，聚焦关键模块？  “乘法”（扩展协同能力）：看门狗能否与其他系统（如温控模块、电池管理）联动，形成异常处理链？  “加法”（增强可靠性）：除复位外，是否可添加多级回滚策略（如先尝试释放资源，再重启应用）？  发起人 ：组长本组自发 lable  <递进问题issue27:提升>在多种运行模式共存的嵌入式控制系统中，模式切换过程中是否可能存在冲突或转场代价？系统是否具备有效的状态缓冲机制与切换确认逻辑，以保障切换过程的稳定性与安全性？  发起人 ：组长本组自发 lable  <递进问题issue28:融合>在引入“外部中断唤醒”（如按键触发或传感器事件）的扩展设计中，是否会显著增加休眠机制的系统复杂度？在实现响应灵敏的同时，如何权衡模块扩展带来的资源消耗与整体系统的稳定性？  发起人 ：组长本组自发 lable  <递进问题issue29:提升>如何通过软硬件协同优化实现智能车多级功耗管理与实时性的平衡？  发起人 ：组长本组自发 lable  <递进问题issue30:提升>在车联组队场景中，如何设计安全可靠的跨车数据交互机制以防范非法入侵？  发起人 ：组长本组自发 lable  三、【代码注释改进工单Board】聚焦成任务工单  1。layout 认领/指定负责人：需求员-郭子涵  2。drawable 认领/指定负责人：周志员-武宇航  3。**Blacktest.java和car.java** 认领/指定负责人：组长助理-何叶  4。**Cardetail.java和carteam.java** 认领/指定负责人：优裁员-王晶 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问** | **附件参考材料** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | 四、【具体版本设定】  原有版本命名：V1.X（基于浅度课设的双 UI 点名功能，聚焦基础硬件控制）  核心功能：  实现蜂鸣器群组点名，数码管和点阵显示被点名者信息（如 PPT 中 “V1.0 蜂鸣器群组点名，数码管显示被点名者信息”）。  中级 UI：通过拨码开关、数码管、点阵、蜂鸣器完成点名操作；高级 UI：引入触摸屏交互，支持单车辆状态显示。  功能局限：仅支持单一点名，无车队协同或智能控制逻辑  改进后版本：V1.X+1（深化双 UI 解耦，融入车联主题元素，向深度课设过渡）  核心升级点：  双 UI 独立解耦：  中级 UI（最简模式）：完全通过硬件接口操作（拨码开关选择车辆、按键触发点名），点阵仅显示车辆编号（如 “M01” 表示成员车 1），蜂鸣器单次短鸣确认操作。  高级 UI（正常模式）：触摸屏独立控制，界面新增 “车联 I・Eo-Smart” 主题背景，支持多车辆图标显示，点名结果以浮窗形式动态反馈（如 PPT 中 “群组点名 UI” 截图）。  车联元素初步植入：  高级 UI 加入 “车辆激活状态” 标签（如 “活跃 / 休眠”），对应实验箱 LED 灯状态（LED1 亮→车辆 1 活跃）。  新增 “头车标识” 功能：组长车点名时，点阵显示 “C” 字母 + 编号（如 “C01”），成员车显示 “M” 字母 + 编号（如 “M02”），强化团队层级逻辑（参考 PPT 中 “组员介绍” 的组长角色设定）。  操作逻辑优化：  双 UI 切换时自动同步状态：从高级 UI 切至中级 UI，保留当前激活车辆信息（如拨码开关自动对应触摸屏勾选的车辆）；切回后恢复界面状态。  新增 “点名超时机制”：中级 UI 中若 30 秒未响应，蜂鸣器长鸣报警，高级 UI 弹出 “超时重试” 按钮（参考 PPT 中 “看门狗机制” 的容错设计思路）。  版本递进价值：  承上：延续浅度课设的双 UI 框架，确保点名功能的稳定性与硬件兼容性。  启下：通过主题化 UI 设计（如 PPT 中 “个性化 Launcher” 元素）和车联角色区分，为深度课设的车队管理、能量共享等功能奠定交互基础，体现 “除（简化冗余）→乘（扩展场景）” 的三法则升级逻辑。  五、【递进目标】  双 UI 解耦（独立中级 UI、独立高级 UI、混合点名）  通过分离中级 UI 与高级 UI 的交互逻辑和硬件资源，实现三种模式的灵活切换与协同，提升系统的可维护性和场景适应性。  1. 独立中级 UI（拨码开关点名，大屏清空）  交互逻辑：  完全硬件驱动：仅通过 8 位拨码开关选择被点名车辆（如拨码开关 1-7 对应车辆 1-7），按下数码管下方 “点名键” 触发蜂鸣器响应，16×16 点阵显示车辆编号（如 “M03”）。  大屏清空机制：点名时实验箱屏幕仅显示基础状态（如温度、时间），清除高级 UI 的触屏交互界面，避免操作冲突。  技术实现：  关闭 Android 系统的触屏驱动，仅保留数码管、点阵、蜂鸣器的硬件控制线程。  通过 GPIO 中断监听拨码开关状态变化，触发点名逻辑（参考 PPT 中 “硬件裁剪” 思路，简化非必要功能）。  2. 独立高级 UI（触摸点名）  交互逻辑：  纯软件交互：在 Android 界面通过触摸屏选择车辆（列表勾选或图标点击），点击 “开始点名” 按钮触发动画效果（如车辆图标高亮 + 蜂鸣器短鸣），点名结果在屏幕底部以标签形式显示（如 “车辆 2 已激活”）。  数据可视化：同步显示车辆实时状态（电量、速度），支持滑动切换车辆详情（参考 PPT 中 “车辆信息显示窗口” 设计）。  技术实现：  使用 RecyclerView 展示车辆列表，通过OnClickListener处理触摸事件。  与硬件解耦：点名指令通过软件模拟（不依赖拨码开关），仅通过蜂鸣器和 LED 灯提供反馈（如 LED2 亮→车辆 2 激活）。  3. 混合点名（双 UI 协同）  交互逻辑：  跨 UI 操作：  中级 UI 触发点名（如拨码开关选择车辆 3 + 按键点名），高级 UI 同步更新状态（车辆 3 图标显示 “活跃”）。  高级 UI 批量选择车辆（如勾选车辆 1、4），通过 “同步至硬件” 按钮，自动设置对应拨码开关状态（开关 1 和 4 拨至 On）。  协同场景：  硬件优先模式：当拨码开关与触屏选择冲突时（如拨码开关 2-On 但触屏取消勾选车辆 2），以硬件状态为准，高级 UI 弹出提示 “硬件状态已变更”。  混合控制模式：通过中级 UI 完成单点故障排查，同时通过高级 UI 监控整体车队状态（如多车点名时，中级 UI 逐一点名，高级 UI 汇总显示所有车辆响应结果）。  技术实现：  建立软件与硬件的状态映射表（如carStateMap），实时同步双方数据。  使用Handler实现跨线程通信，确保硬件操作（如拨码开关）与 UI 更新（如屏幕刷新）的时序一致性（参考 PPT 中 “GitLab 版本控制” 对状态同步的管理思路）。  4. 递进目标价值  解耦优势：  中级 UI 专注硬件可靠性，高级 UI 专注用户体验，降低模块间耦合度，便于独立调试（如单独测试触屏逻辑或硬件驱动）。  符合 “三法则” 中的 “除” 法则：裁剪冗余交互，确保基础功能（点名）在不同环境下的稳定性。  场景覆盖：  工业场景：通过中级 UI 快速完成硬件故障定位（如单车辆点名测试）。  日常场景：通过高级 UI 实现批量点名和状态监控，提升操作效率（如车队出发前的快速点名）。  技术储备：  为深度课设的 “车队协同” 功能奠定基础（如混合点名模式可扩展为 “头车通过触屏控制，成员车通过硬件响应”），体现从 “浅度课设→深度课设” 的平滑过渡（参考 PPT 中 “流程分解” 对多阶段协同的设计）。  六、【修正完善本组浅度课设需求&验收单】  浅度课设需求修正与补充：在原有双 UI 点名功能基础上，中级 UI 调整为仅通过拨码开关、数码管、点阵及蜂鸣器实现单车辆点名操作，新增 “头车优先点名” 逻辑（组长车点名时蜂鸣器长鸣 + 点阵闪烁）；高级 UI 引入触摸屏独立点名，界面分区显示车辆编号、状态（活跃 / 休眠）及小组 LOGO（如 “NK2025 车联 I・第二组” 标识），按钮样式统一为车联主题图标（如方向盘、电池）。多阶段系统的启动阶段增加硬件自检功能（检测数码管、蜂鸣器状态，异常时弹出警告），点名阶段支持双 UI 切换并记录响应时间。硬件与软件联动方面，新增混合模式下的状态同步机制，即中级 UI 点名后高级 UI 自动标记车辆为 “活跃” 并高亮显示，反之亦然。  验收单（浅度课设阶段）：功能完整性需验证中级 UI 与高级 UI 的独立点名操作，确保硬件响应（数码管显示、蜂鸣器发声、LED 灯点亮）与软件反馈（界面状态更新）一致；双 UI 解耦要求两类界面操作互不干扰（如中级 UI 操作时高级 UI 界面清空），仅在混合模式下允许状态同步；数据同步需测试双 UI 间点名结果的一致性（如中级 UI 点名车辆 3 后高级 UI 同步显示其活跃状态）；用户体验方面，高级 UI 需符合车联主题布局且操作流程不超过 2 级，中级 UI 响应延迟需控制在 0.5 秒内；文档与协作需检查代码注释的小组标识（如 “// 第二组 - 金莫迪”）及 GitLab 分支管理记录。  与深度课设的衔接：浅度课设的双 UI 解耦和混合点名机制为深度课设的车队协同功能提供底层支持，例如深度课设中可扩展为组长通过高级 UI 发起车队批量点名、成员车通过中级 UI 独立响应，同时浅度课设的状态同步逻辑（如CarState类管理）可直接应用于深度课设的多车状态协同，实现 “深浅课设流畅对接” 的递进目标。 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问** | **附件参考材料** | **展示** |
|  | 附件二 课程三栏认知OKR建议 **通过嵌入式系统授课讲解、浅度群组点名课设熟悉与四主题课设深度迁移体验，**  **在深入学习微观节点五层结构普适专用定制计算相关知识的同时，**  **SAMRT（具体·可抵达·可量化·相关性·时间边界）原则目标设定！！**  **向上课程对齐O[0]熟练运用五层结构，构造初步嵌入式系统**  **向上课程对齐O[1]运用三栏OKR与裁剪先进工具能力，**  **向上课程对齐O[2]完成三法则嵌入分析与目标指向递进，**  **向上课程对齐O [3]理解宏观嵌入式思维的消创数据生态。·**  **向上课程对齐O [4]打下计算机相关专业定制计算基础，**  **向上课程对齐O全面打开放开公开自我生命系统，示范群组主动、合作与探索性学习，**  **向上课程对齐O以计算定制培养诠释践行·南开公能校训。**  **左右课设对齐：深浅课设流畅对接·**  **左右课设对齐：不同主题XXX除本元对比·**  **左右课设对齐：不同主题XXX乘放大效能对比·**  **左右课设对齐：不同主题XXX加回滚合理自然比对·**  **左右课设对齐：不同角色不同阶段的XXX长板对齐**  **左右课设对齐：结伴小组XXX示范对齐**  **左右课设对齐：小版本递进中的精进闭环工具对齐**   1. 组员student-OKR   ****目标****：通过浅度到深度课设的递进实践，掌握嵌入式系统开发全流程技能，实现从代码编写到系统优化的能力跃升，同时强化团队协作与问题解决意识。 ****O1：掌握嵌入式基础开发流程与核心技能**** ****KR1****：熟练使用 Java 语言完成浅度课设双 UI 点名功能（中级 UI：拨码开关 + 数码管；高级 UI：触摸屏），确保代码注释覆盖率≥80%，并通过 GitLab 提交至少 5 次有效代码迭代。  ****KR2****：在深度课设中，独立完成 1 个硬件模块控制（如 LED 灯状态显示、AD 电位器温度采集），实现模块驱动代码与应用层逻辑的解耦，代码通过率≥90%。  ****KR3****：理解椭圆曲线离散对数问题（ECDLP）的基本原理，能复现简单差分分析案例（如文档中 4 位分组密码示例），并撰写 500 字以上分析笔记。 ****O2：提升系统设计与优化能力**** ****KR1****：在主题化设计（车联 Ⅰ）中，参与完成 1 项核心功能开发（如结伴运行逻辑、变姿控制算法），确保功能在实验箱 Android 平台运行稳定，UI 响应延迟≤500ms。  ****KR2****：运用 “除、乘、加” 三法则优化系统：  ****除法****：简化非核心交互流程（如合并冗余按钮），使操作逻辑深度≤2 级；  ****乘法****：放大核心场景体验（如组队成功时增加语音提示），用户反馈满意度≥80%；  ****加法****：扩展异常处理机制（如看门狗触发时保存行驶数据），系统故障率降低 50%。  ****KR3****：参与功耗优化任务，实现最简模式待机电流≤0.5A，正常模式平均电流≤0.75A，通过示波器实测并记录数据。 ****O3：强化团队协作与问题解决能力**** ****KR1****：在 GitLab 中主动认领至少 3 个 Issue（如 “UI 切换防抖处理”“多车入队冲突仲裁”），定期更新任务进度，与组员协作解决技术难题。  ****KR2****：在组内技术讨论中，每周提出至少 1 个有效改进建议（如代码优化方案、测试用例设计），并通过 Issue 评论或会议记录留痕。  ****KR3****：参与跨组技术交流，学习其他主题（如纸联、体联）的优秀实践（如传感器数据处理、UI 动画设计），输出 200 字以上学习总结。 ****O4：完成文档撰写与成果展示**** ****KR1****：独立撰写课设报告中的 “模块实现细节” 章节（≥1000 字），内容包含代码逻辑图、测试数据表格及问题解决过程。  ****KR2****：协助制作展示 PPT，负责 “技术难点与创新点” 板块（如差分分析在车联安全中的应用设想），确保 PPT 内容与代码实现一致。  ****KR3****：在成果展示中，清晰演示负责模块的运行效果（如变姿控制的 LED 灯状态变化），并准确回答评审提问。  **具体到组内个人：**  组长：巩岱松  目标（O）：  确保项目按时完成，达成设计目标，提升团队协作效率。  关键结果（KR）：  KR1：制定详细的项目计划并监督执行，确保按时完成所有任务。  KR2：通过定期会议和沟通机制，确保团队成员之间的信息流通顺畅。  KR3：成功实现EO-Smart控制板的核心功能，包括电量显示、休眠功能等。  KR4：推动团队创新，提出并实现至少两项系统优化或功能改进。  周志员：武宇航  目标（O）：  优化代码管理，提升团队的开发效率和代码质量。  关键结果（KR）：  KR1：通过GitLab实现代码的有效版本控制，确保代码库整洁。  KR2：每周至少完成一次代码审查，及时发现并解决潜在问题。  KR3：协助其他成员解决技术难题，提升团队整体技术水平。  KR4：完成至少两个模块的代码优化，提高系统性能。  优裁员：王晶  目标（O）：  优化硬件软件资源，提升系统性能和用户体验。  关键结果（KR）：  KR1：完成硬件资源的裁剪，减少不必要的功耗和资源占用。  KR2：优化软件代码，提高系统的运行效率和响应速度。  KR3：设计并实现用户友好的UI界面，提升用户操作体验。  KR4：完成至少两个功能模块的优化，确保系统稳定运行。  展示员：金莫迪  目标（O）：  编写代码，高质量完成项目展示，确保项目成果得到充分展示。  关键结果（KR）：  KR1：制作高质量的项目展示PPT，清晰展示项目成果和创新点。  KR2：拍摄并剪辑项目演示视频，突出项目亮点和操作流程。  KR3：在项目汇报中清晰、准确地讲解项目内容，回答评委和观众的问题。  KR4：协助其他成员完成代码编写和功能测试，确保项目顺利进行。  需求员：郭子涵  目标（O）：  准确把握用户需求，确保项目功能满足用户期望。  关键结果（KR）：  KR1：完成详细的需求文档，明确项目功能和用户需求。  KR2：与团队成员密切合作，确保需求得到准确实现。  KR3：收集并整理用户反馈，为项目优化提供依据。  KR4：完成至少两个功能模块的需求分析和设计，确保功能符合用户需求。  组助：何叶  目标（O）：  协助组长工作，促进团队合作，确保项目顺利进行。  关键结果（KR）：  KR1：协助组长制定项目计划和任务分配，确保团队成员明确职责。  KR2：协调团队成员之间的沟通和协作，解决团队内部的矛盾和问题。  KR3：完成组长交待的其他任务，确保项目顺利推进。  KR4：协助其他成员完成任务，提升团队整体工作效率。 |  |

|  |  |  |
| --- | --- | --- |
| **发问** | **附件参考材料** | **展示** |
|  | 附件三：学习认知节律 |  |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | 1. 6~8点：记忆复杂公式定理或单词   清晨6点是身体休息完毕、肝脏排毒结束、身体进入如兴奋状态的时候。此时大脑清醒，记忆力强，进入一天中第一次最佳记忆期，我们可以安排复杂定理、公式的记忆或者背诵记忆。   1. 8~9点：攻坚克难理科类学习   8~9点是人体精力旺盛，神经兴奋度提高，且记忆仍保持最佳状态，这段时间我们可以安排需要精确演算、高强度思考，周密判断等难度大、需要攻坚的内容，如进行数学、物理、化学等学科的学习。   1. 9~11点：复习、短期记忆最佳时机   这段时间身心处于积极状态，记忆力突出，短期记忆最佳时间，创造力最旺盛时刻，可安排复习外语、语文、历史、政治文科等;   1. 13~14点：该休息了   这段时间一般处于饭后，第一阶段的兴奋期已过，精力消退，进入低潮阶段，此时反应迟缓，容易疲劳，可安排适当休息，为下一阶段的兴奋期蓄力。   1. 15~16点：阅、诵、背时间到。经过午休后，身体状态重新改善，感觉器官此时尤其敏感，精神抖擞，科学试验表明，此时长期记忆效果非常好，可合理安排需“永久记忆”的内容，如阅读、听力、诵读、默写等内容。 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |
| **发问** | **附件参考材料** | **展示** |
| **操作发问**  **编程发问**  **任务发问**  **递进发问**  **关联发问** | 1. 17~18点：解决烧脑问题专属时段   这段时间体力、耐力达一天中的最高峰，试验表明：这段时间是完成复杂计算或比较耗脑作业的好时期。可安排数学、物理、化学等计算分析题目的练习;   1. 18~22点：难易交替，文理交替的学习   其中：19~20点，随着体内能量消耗，情绪不稳,可安排短暂休息。  其中：20~22点，这段时间大脑又开始活跃，反应迅速，记忆力回升到不错状态，直到睡前为一天中最佳的记忆时期。所以可以安排需要记忆的知识进行学习。  22~24点，细胞修复工作开始，可以开始安排休息了。 | **结果展示**  **问题展示**  **困难展示**  **反省展示**  **合作共生展示** |