

嵌入式打点问题

车联I 第二组 组长巩岱松

1 🌟 问题一——看了种子判断：

1.1 TODO-C1：智能车支持哪些“姿态”？何种情况下需要改变？如何模拟变姿？

任务核心：识别并设计出EO-Smart的多种“变姿”模式，并在不同场景中切换姿态。

结合三法则分析：

1. **除法分析：**强调“短途联通”与“能量播撒”的轻量性与灵活性，需要不同姿态适应不同场景（如：停车、掉头、进队、侧向移动等）。
2. **乘法分析：**变姿能力可增强交通适应性（如折叠、剪刀门、轮组90°旋转等）。
3. **加法分析：**基于“系统韧性”需要最小化回滚姿态作为基础原型。

模拟方式：通过 `Toast` 提示文本、`MediaPlayer` 播放提示音、展示 `图片/动画资源` 等形式可视化不同姿态切换效果。

1.2 TODO-C2：车队应使用什么数据结构管理？如何模拟出队与入队？

任务核心：设计车队结构和模拟车辆的加入/离开过程。

结合三法则分析：

1. **除法分析：**短途出行与能量播撒强调协同队列。
2. **乘法分析：**群组协同驾驶要求灵活的队列组织与数据共享。
3. **加法分析：**进出队控制需基于能源预测、位置管理。

推荐数据结构：使用队列（Queue）+ 哈希表（HashMap）组合：队列用于表示车队的先入先出结构；哈希表记录每辆车的唯一信息（如车辆ID、能源状态等）。

模拟方式：按钮控制 + 动画列表更新 + 日志文本展示。

1.3 TODO-C3：如何实现深度课设→浅度课设的主题“回滚”机制？

任务核心：支持从高智能（群联车）到普通模式的回退。

结合三法则分析：

1. **除法分析：**回滚至最小功能支持老年人“傻瓜操作”。
2. **乘法分析：**功能解耦支持多阶段智能实现（AI辅助→自动驾驶→群联）；
3. **加法分析：**系统“韧性设计”强调可回滚能力。

模拟方式：设置状态枚举（群联车→自联车→普通车）；提供回退操作按钮，结合视觉反馈实现状态切换；可视化状态树或线性状态切换流程。

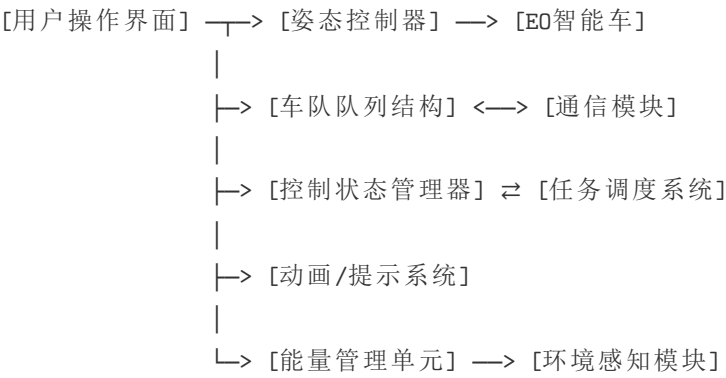
2 🌟 **问题二——理解种子判断：**

C1&C2任务中明确涉及的实体对象：

实体对象	属性（举例）	功能描述
智能车（EO 车）	车ID、姿态状态（折叠/展开/旋转）、能源状态、电量	系统核心单位，支持姿态变换与入队出队
姿态控制器	当前姿态、目标姿态、变换逻辑	接收输入并驱动车体姿态变化
用户操作界面	按钮事件、状态提示（Toast/动画/语音）	提供姿态切换和队列控制的人机交互入口
车队队列结构	队列顺序、成员列表（车辆ID）	维护当前车队结构、支持入队/出队操作
能量管理单元	电量值、能耗模型、能量共享模式	决定车辆是否能变姿、是否允许加入车队

建议补充的隐含关键实体对象：

实体对象	属性或行为	作用说明
控制状态管理器	当前系统模式（普通/自联/群联）、状态栈	管理模式状态变化与回滚（C3任务关键）
通信模块（V2V）	车辆ID、通信协议、队列状态	实现车与车之间信息共享和同步协作
环境感知模块	激光雷达/摄像头数据、障碍物识别	决定姿态变换是否可执行、路径是否合理
动画/提示系统	Toast文本、动画资源、语音提示	用于可视化反馈交互（“模拟”变姿等）
任务调度系统	当前任务列表、执行队列、优先级	管理异步任务，如变姿执行、入队出队处理
开发与调试接口模块	日志输出、状态快照、调试按钮	支持开发中调试与状态跟踪



2.4 如何在任务中使用这些对象：

- 前端任务（UI）主要和：用户操作界面、动画提示系统交互；
- 后端控制逻辑主要由：姿态控制器、车队队列结构、状态管理器 和 通信模块 构成；
- 系统支持层包含：环境感知模块、能量管理单元、调度系统等底层服务；
- 状态切换（C3）：通过控制状态管理器管理不同模式间的转换（如回滚至普通车）

3 问题三 —— 目标形成判断：

3.5 “除乘加” 三法则对应设计理解

1.除：拆分功能、识别任务边界：宏观层面：EoSmart作为智能出行终端，拆解为“姿态控制”、“自动驾驶”、“车队协作”、“能耗播撒”等子系统。微观层面：各功能对应五层结构中不同模块，有助于从整体架构中理清职责分工与调用流程。

对应结构：硬件层 → 引导层 → 内核层

2.乘：融合能力、叠加模块价值：姿态切换 + 自动感知 + ai辅助驾驶等功能联合，构建可组合、可扩展的车体控制生态。

对应结构：内核层 → UI层 → 应用层

3.加：强化体验、增强系统弹性：增加人机交互方式（图像、声音、文本），引入任务回滚机制，支持不同层级的控制回退（群联→自联→普通）。

对应结构：UI层 → 应用层（含用户体验与逻辑可视化）

3.6 嵌入式系统五层结构与EoSmart框架对齐关系

嵌入式五层结构	EoSmart体现	设计目标涌现
硬件层	电机、传感器、激光雷达等底层组件	提供执行动作与数据采集能力
引导层	系统引导程序、Bootloader、设备初始化	启动设备、加载内核配置
内核层	姿态算法控制、队列调度、车间通信逻辑	管理任务调度和通信协议
UI层	Toast提示、动画演示、语音反馈等交互方式	实现人性化交互，提升用户感知
应用层	功能整合与界面配置，任务切换机制	实现高层逻辑调度与系统弹性切换

3.7 当前阶段“目标对齐”的典型涌现内容

- 姿态模拟：由内核层驱动硬件执行，UI层展示结果，应用层支持切换。
- 队列入队出队：由内核层调度，UI层呈现队列状态，硬件层展示移动效果。
- 主题回滚机制：通过应用层逻辑控制向下协调内核与引导层，模拟功能等级回退。

4 🌟 问题四——任务分解判断

4.8 C1：姿态变换任务

前端任务：设计按钮和动画界面，展示车辆“折叠、展开、转向”等姿态变化；可通过文本提示（Toast）、音效（MediaPlayer）、动画（ObjectAnimator/Lottie）模拟变姿过程。

后端任务：建立姿态状态管理模块（如使用 `enum`），接收前端指令，判断当前是否允许变姿，并向前端发出变姿控制信号。

4.9 C2：车队队列管理任务

前端任务：开发车队展示界面，实现入队/出队按钮与动画效果，动态可视化车队变化。

后端任务：采用 `Queue` 或 `Deque` 结构管理车辆队列，实现车辆的入队、出队逻辑，提供接口供前端调用并通知界面更新。

4.10 并行开发建议

- 1.前端负责界面交互与视觉展示，后端负责状态逻辑与数据结构，实现通过接口通信；
- 2.任务可分为姿态模块开发、队列管理模块、控制接口模块三部分，分别由不同成员承担；
- 3.每个模块提供统一的调用接口，便于独立开发、集成测试。

5 🌟 问题五——分工协作判断

在上线版本的开发过程中，为实现码释量化周120行，大版本500行的目标，小组成员需明确分工明确分工，每人每周负责20~30行独立模块，保证开发量。对于大版本的开发围绕C1/C2/C3的核心逻辑拆分功能，按阶段集成，每周形成一次小迭代。各角色协同合作完成公分精进私分：

角色	职责重点
组长	统筹进度，分配任务，主导版本集成与冲突解决，协调团队对齐目标。
需求元	跟进除乘加分析与用户视角，细化任务需求，监督是否贴合课设目标。
周志员	每日记录进展，定期输出周报、问题列表，推动持续改进。
优裁员	审查代码质量与接口一致性，确保开发文档与逻辑规范，进行版本评审。
展示员	负责动画演示、变姿可视化效果制作，撰写讲解稿并组织模拟讲解。
组助	支持测试、日志管理、资料整理等工作，补位其他角色的临时任务空缺。

全体成员共同努力完成向深度课设的进展项目内嵌清晰注释，便于组内交叉理解与任务衔接；实现“项目即教材、开发即授课”的循环式成长机制！