**NK2025嵌入式系统课程设计小组个人报告**

课设主题名： **车联 I：EO-Smart老年代步（T·I）电动短途物联代步车**

【报告摘要】

**组号： G2 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 组助：何叶**

**个人公共角色贡献：**作为小组的需求员，我始终以推动团队需求梳理和功能落地为核心，积极参与每周课设目标的需求细化。每周牵头“实验规划表”和“三栏需求”文档的编写与更新，确保团队成员对任务需求和目标规划等有清晰认知。针对系统核心功能（如车辆多状态切换、功耗调度、看门狗机制等），我负责需求逻辑的抽象与分解，协助组长制定任务路径图，及时同步需求变更，梳理课设项目的“除乘加”三法则。在团队协作中，我主动收集组员在开发过程中遇到的需求疑问，转化为可量化的issue，推动问题的快速反馈与闭环。提出创新性需求建议，协助展示员完善课设总结材料，提升团队整体的需求管理与文档规范水平。

**个人本分贡献：**作为需求员，我负责课设实验各阶段的需求梳理与功能验收，主导实验规划表”和“三栏需求”文档的撰写和维护，确保每周8XML文档内容的及时、准确填写。针对高级休眠、模式控制、能量共享等核心功能，深入参与对象抽象、结构分解与目标定义，推动需求与代码实现的高效对接。在GitLab平台上，我每周及时反馈需求变更及任务分配情况。协助组长和开发成员完成Launcher界面优化、车辆姿态评估系统和参与群组点名等关键模块的需求分析与代码编写。

**个人自评： 0.90，** 在担任需求员期间，能够结合课程三法则理念，持续推进需求梳理与文档规范，确保团队成员对课设目标和功能实现有清晰认知。积极参与需求创新与问题反馈，推动团队高效协作和成果落地。通过多轮需求优化，提升了系统的功能完整性和用户体验，对团队整体进度和质量做出积极贡献。

**组长互评：0.92，**需求员在项目需求梳理、文档编写和任务分解方面表现突出，能够及时发现并反馈需求问题，推动团队高效协作。对课设目标和功能实现有深入理解，并提出多项创新建议，参与到项目的核心功能的代码编写，为团队成果的完善和展示提供了有力支持。

**报告个人自述比例：85%**

目 录

一、主题嵌入对象最简调研（建议400字）

二、主题嵌入式三法则宏微分析（建议300字）

三、主控硬件与I/O配置理由（建议300字）

四、最小软环境定制（**建议250**）

五、浅度与深度应用系统设计（建议500字）

六、主题深度应用系统Vibe编程（建议500字）

七、可测与可优设计（建议250字）

八、可靠性与性能优化关联（建议200字）

九、系统性能量化小结（建议**150字**）

十、多级目标牵引/问题链与版本控制工具使用心得（**建议250字**）

十一、小组与个人工作总结与问题反省（**建议150字·展示课后完成**）

本人手写签名：

2025年6月19日

1. **主题嵌入对象最简调研** 建议400字（参照前期小组调研，小组内共享）

**1.行业基本状态: 建议200字**

**简单历史：**

轻量级智能交通的发展始于21世纪初，随着城市交通拥堵与环保压力的上升，微型电动车逐渐成为替代传统私家车的解决方案。EO智能连接车作为概念验证平台，自2011年起在德国展开研发，集成变姿、感知、自主控制等前沿技术，探索城市出行的新形态。

**主要危机与趋势：**

当前行业主要面临集成复杂度高、续航能力有限、成本控制困难、车联协议缺乏统一等问题。发展趋势正向模块化、小型化、高度智能化推进，特别重视面向老年人、弱势群体的安全设计和“低能耗+高可靠”系统构建，以及群组协同与车联网集成能力的提升。

**2.整体框架: 建议100字**

**主要元素：**

系统包含四轮轮发电机、线性执行器变形底盘、视觉与激光雷达感知模组、多车协同通信结构与自动导航控制模块。

**基本原理：**

通过执行器调节车体姿态和尺寸，结合多模态传感器进行环境识别，再以局部自主决策配合群组互联，实现可变形、可协同、可降阶的智能电动车运行框架。

**3.对象属性: 建议50字**

具备折叠变形、可侧向移动、多点感知与能量共享能力的模块化微型电动车平台，适用于复杂城市环境与短途高频出行。

**4.设计聚焦: 建议50字**

专注于“可变姿态+能量共享+群组协同”，提升老年用户智能短途出行的便捷性、安全性与交互友好性，构建柔性可退化的车联系统。

**二、主题嵌入式三法则宏微分析** 建议300字

**1.宏微·洞察除法 建议80字（参照授课四主题分析总表及规划表）**

**宏观：**聚焦老年人短途出行痛点，通过精简操作层级与冗余功能，降低技术使用门槛。以“银发友好路径”为核心，集成一键呼救、目的地预设等必需功能，填补“最后一公里”服务盲区。

**微观：**

1.短途移动联通：在灵活轻便的自动化基础上，自动操控系统可实现车队互联与结伴出行，支持车辆入队、出队等多种操作流程

2.移动能量播撒操控：系统可在出行过程中实现能量的自动流转，用户仅需通过简单按键便可轻松控制车辆

**2. 宏微·增强乘法建议 建议80字（参照授课四主题分析总表及规划表）**

**宏观：**通过模块化设计与动态协同能力的指数级叠加，形成“一车多用、一能多效”的复合价值网络。不仅提升车辆自身的适应性，更通过系统级优化重构交通资源分配规则。

**微观：**

1.数据×激励：头车达加速条件时，自动同步激发后车响应，车队速度倍增。

2.形态×能耗：翘臀态降低风阻时，自动启动车队电量均衡功能，延长续航15%。

3.模拟×安全：震动/旋转动画还原真实路况 × 7%电量双级制动（先减速后停车）

**3. 宏微·包容加法 建议60字（参照授课四主题分析总表及规划表）**

**宏观：**以“回滚基座+能耗中台+协同顶层”构建阶梯式演进架构。从单体操控到车队协同，最终实现自动驾驶，逐层叠加安全冗余与能源分配能力。

**微观：**

1.系统回滚机制：最小化回滚形态（如操作中断自动返回主界面） + 电量阈值减速机制 + 能量共享协议，形成三级容错。

2.能耗智能管理机制：结合车队电量控制车队的进出节奏，可实现能量的动态流转与群组高效运行；充电与耗电都可以进行，配备相对应的充电桩，可以使车队便捷的实现相关电量的分配与运行。

3.协同驾驶的生态扩展：基础组队功能 + 自动规划路线+能源分配，逐步构建交通微生态。

**4.目标与主要指标设定 建议80字**

我们实现了一些项目的基本功能：

电量显示，根据电量来对路径进行最佳规划，设定电量阈值判断，触发渐进式减速机制。

出队入队功能，实现了与附近车辆的互联组队；

软看门狗，解决边界异常等问题；

加入路段拥堵状况，同步调节最佳路径；

风阻协同与能量共享机制：通过减少尾部风阻与电量均衡保持车队的整体运行效率与稳定性

引入震动、旋转等图片动画处理手段真实还原车辆状态；

采用迪杰斯特拉算法进行路径的动态规划

**5.其他：**无。

**三、主控硬件与I/O配置理由**（层一） 建议300字（参照规划表，及课设实验箱参考文档，该部分可以小组共享）

**1.主控硬件规格： 建议50字**

本课设使用基于S5P6818八核ARM Cortex-A53架构的嵌入式开发板作为主控平台。该芯片最大主频可达1.4GHz，具备高性能低功耗特性，特别适合需要处理多任务调度、UI响应与状态感知的嵌入式应用场景。

**2.主控时钟： 建议30字**

主频设置为1.4GHz，结合高速晶振与定时中断管理机制，满足多线程运行与高频刷新率UI渲染需求，有效支撑路径规划与能量管理并行执行。

**3.供电能耗： 建议50字**

系统默认采用5V稳定电源供电，主控平均功耗为250~500mW，采用休眠唤醒机制结合功耗监控模块后，在车队运行状态下提供更长续航能力。

**4.输入配置/模拟： 建议50字**

系统支持多种输入方式：如拨码开关用于配置运行模式，实体按键触发包括启停、异常模拟、状态重置等操作，并通过电阻分压方式模拟真实传感器输入状态，便于测试死锁、路径偏移与UI模式切换效果。设计使用8位拨码开关模拟用户输入，通过设置位值组合模拟功能选择与控制指令；数字按键用于车队ID输入与身份验证；PWR、RESET与BOOT组合键支持强制复位与系统唤醒，触摸屏则用于交互命令选择。借助拨码与按钮，模拟路径切换、姿态调节、共享确认等输入流程。

**5.输出配置/模拟： 建议50字**

输出模块包含LED状态灯（用于表示运行/异常/恢复状态）、蜂鸣器提示系统事件、串口打印日志用于调试反馈，同时提供USB口连接外设测试与BLE模块模拟异构通信。通过输出反馈可完成从状态感知到响应的实时验证闭环。系统通过16×16 LED点阵实现车辆状态动画显示，包含组队完成、路径执行、系统告警等图案提示；8段数码管动态显示温度、电压、电量等数据；蜂鸣器提供突发告警与指令确认提示音；LED双闪灯模拟紧急信号响应。

**6.纸UI描述： 50字**

UI设计采用三层分级结构，分别为极简状态栏（仅展示电量与速度）、中级半屏互动层（集成路径与模式切换）、高阶全屏功能页（包含数据分析、调试与任务回滚），用户可通过边缘滑动手势或语音指令切换，充分体现驾驶场景下注意力分配与交互效率原则。页面划分为三大区域：顶部显示用户身份信息及车队配置情况，中部设置操作命令如“组队”、“解散”、“共享模式切换”等按钮区，底部展示当前路径、姿态模式、电量状态与故障信息，形成可视化透明界面。



1. **其他**

硬件扩展支持UART调试接口、PWM脉宽调制控制、GPIO中断输入等多种外设模块，同时配套Wi-Fi与蓝牙模块，支持拓展组网、通信与远程更新等功能，为系统后续升级与拓展预留充足接口资源。

**四、最小软环境定制（层二&三&四）百字简述 建议250（**参照**自治自学与课设实验）**

**1. u-boot、kernel、Android简介及其之间的关系简述 建议50字**

U-Boot 是嵌入式系统常用的引导程序，负责完成硬件初始化并加载操作系统内核。Linux Kernel 是系统的核心部分，承担设备驱动、资源管理等核心职责。Android 则是基于 Linux 内核构建的操作系统，扩展了用户界面与上层应用功能。三者的关系为：设备上电后由 U-Boot 启动内核，内核运行后加载 Android 系统，从而进入正常工作状态

**2. u-boot编译原理及步骤简述 建议50字**

首先需获取适用于当前硬件平台的 U-Boot 源码，接着根据平台架构增添必要信息并修改配置文件。在 include/configs 中设定目标板级配置，同时在 arch 目录下适配 CPU 架构代码，并更新 boards.cfg 文件以添加新平台。最后执行 make 命令完成编译，生成可执行镜像。整个过程需结合平台特性进行定制化调整。

**3. kernel编译原理及步骤简述 建议50字**

首先从官方或源码仓库下载 Linux 内核源码，进入源码目录后通过 make menuconfig 进行内核参数配置，完成后会生成.config 配置文件。接下来使用 make 命令进行编译，生成内核镜像 zImage 或 Image，最后将生成的镜像文件部署至启动路径以便系统加载使用。

**4. Android编译原理及步骤简述 建议50字**

启动 Android 编译前需初始化环境，指定目标设备与构建方式，并准备如 JDK、NDK、CMake 等依赖。配置完成后，调用 make 工具启动构建流程，对 Android 源码（含系统服务与应用）进行编译，生成 APK 文件及系统镜像。最终可构建 OTA 升级包用于系统更新。

**5. 烧写验证简述 建议 50字**

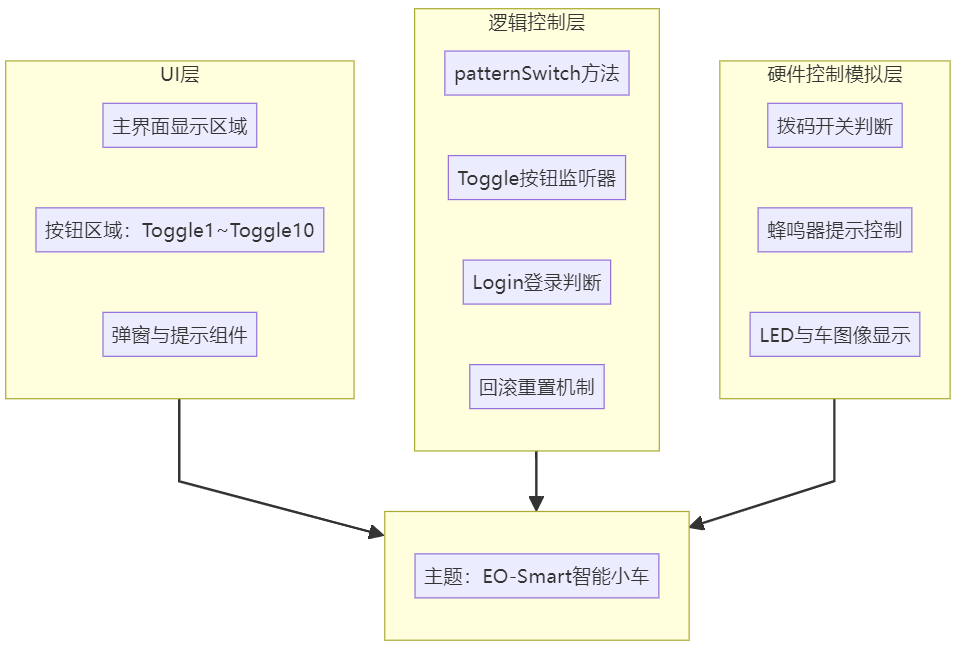
首先需使用工具如 WinImage 对 TF 卡进行分区与格式化，并准备系统镜像文件复制到相应目录。确认目标设备连接无误后上电，即可自动进行烧写过程。烧写完成后，如未发生错误，设备将引导进入系统，实现完整启动流程。

**6.其他：**无

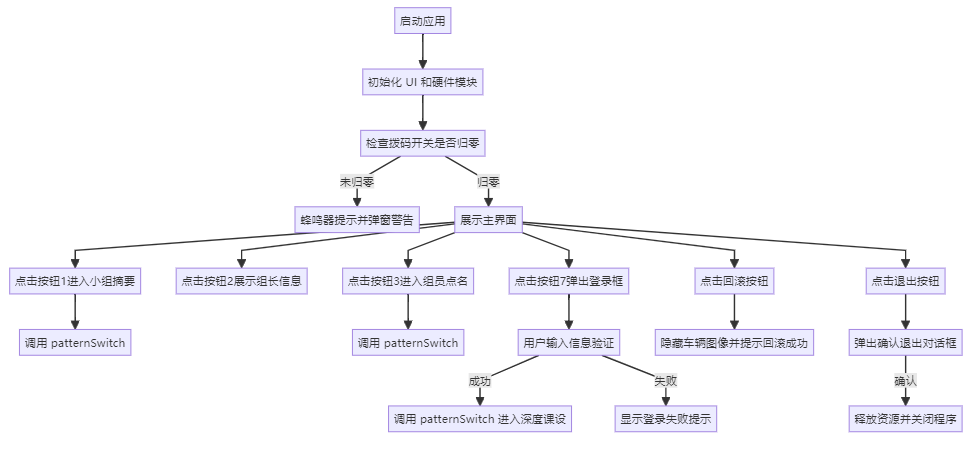
**五、浅度与深度应用系统设计**（层五）简述 建议500字

1.本组浅度课设需求摘要及结构图、流程图、状态图 建议200字

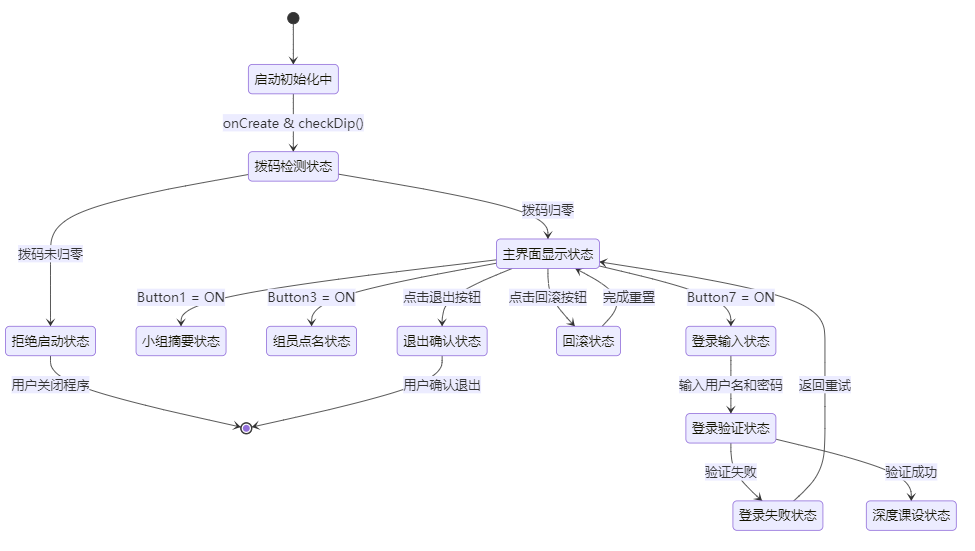
结构图：



流程图：

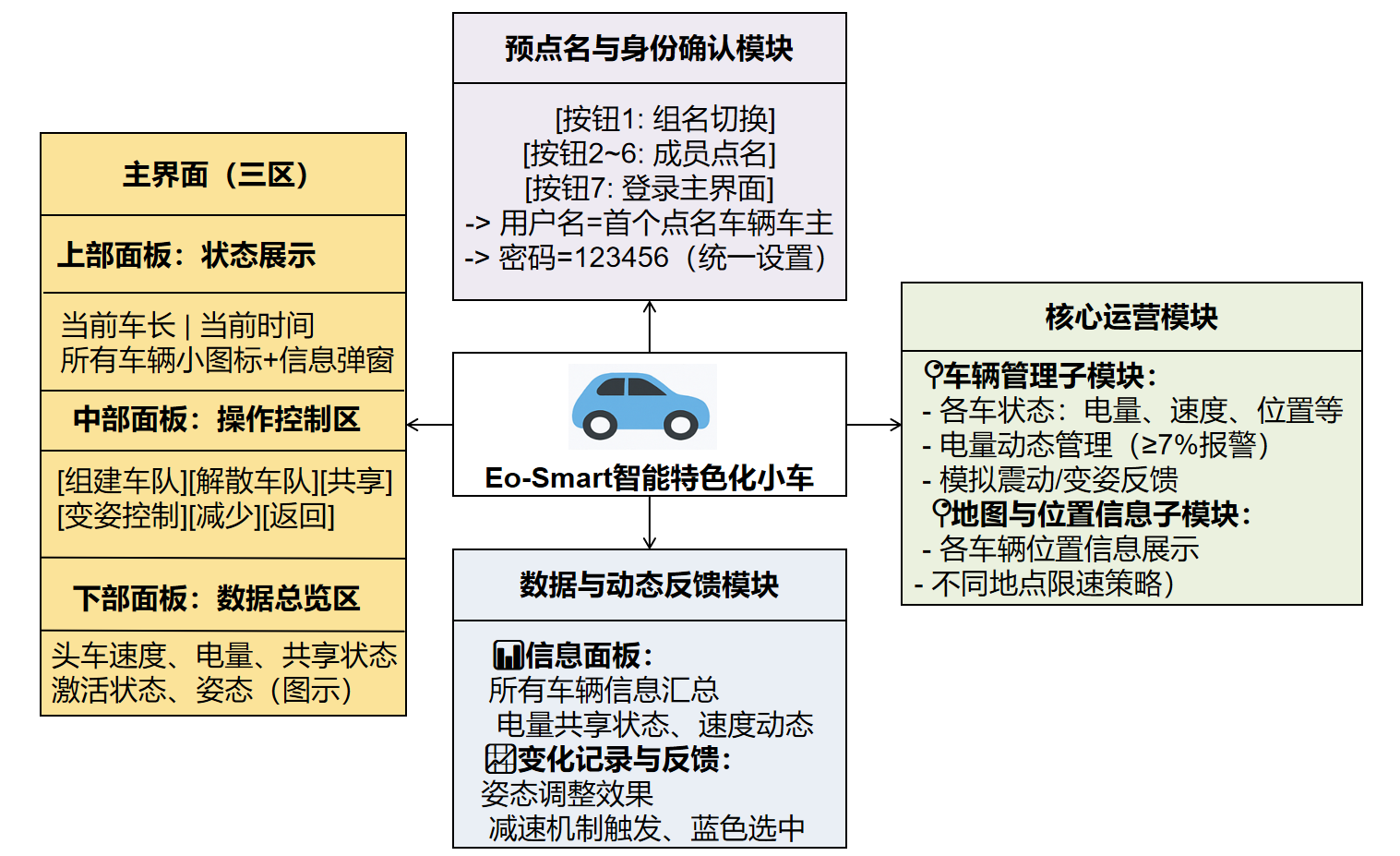


状态图：

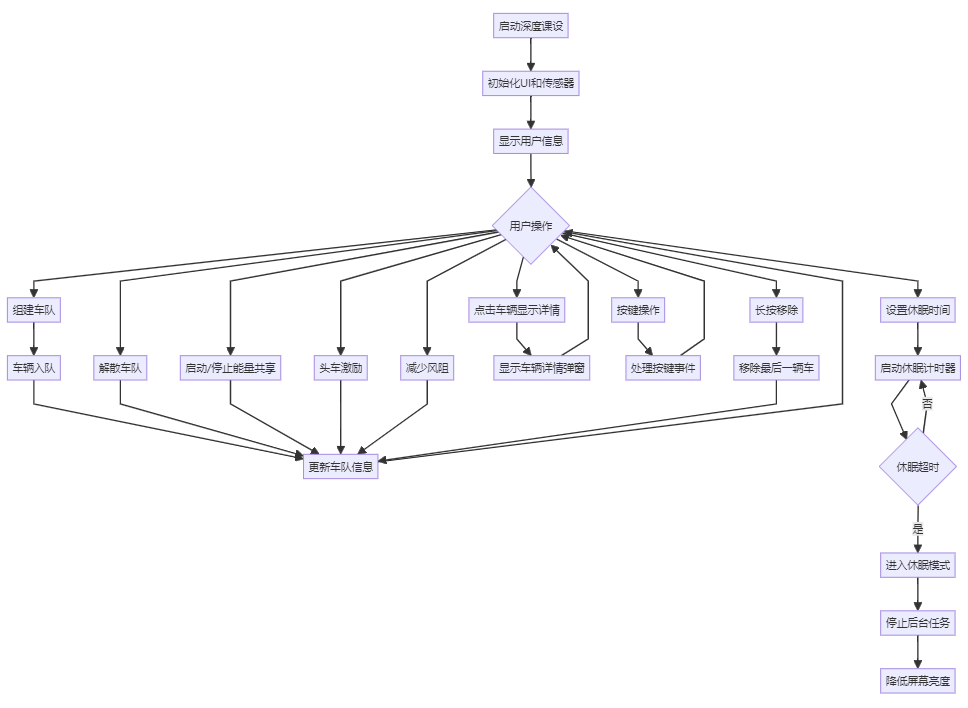


1. 本组深度度课设需求摘要及结构图、流程图、状态图 建议300字

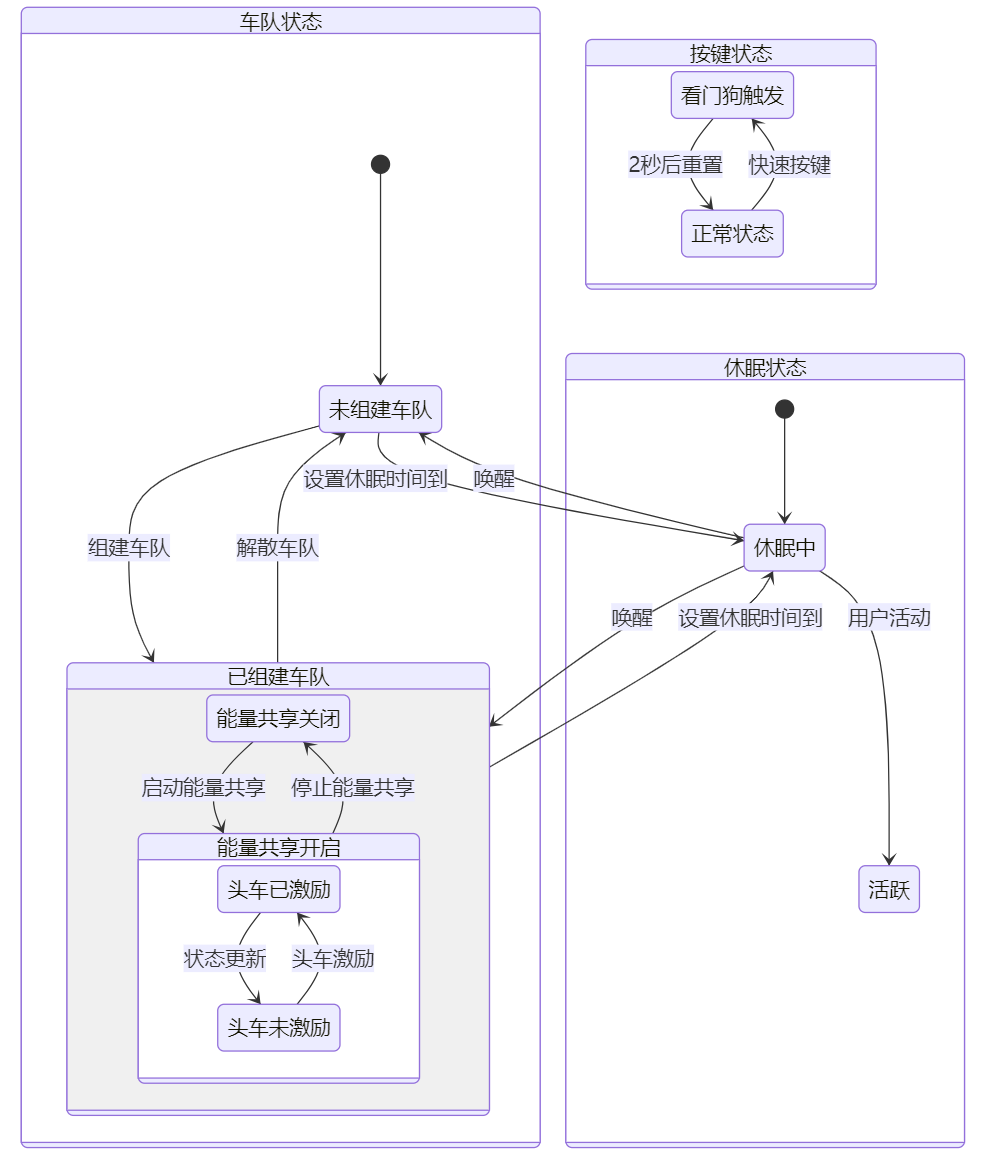
**结构图：**



**流程图：**



状态图：



3.其他

无。

**六、主题深度应用系统VibeCoding编程**（层五） 建议500字

**1.语言编程环境简述 建议50字**

本课设编程环境基于 Java JDK、Android SDK 与 AVD Manager 构建，结合 Eclipse for Android 进行界面与逻辑编程，部分功能模块还涉及 C/C++ 底层控制。同时，引入 VibeCoding 智能辅助模式，通过大模型协同提升代码生成效率与逻辑清晰度。使用Eclipse和试验箱设备完成交互调试，并借助 GitLab 进行小组代码协作管理。

**2.VibeCoding编程底座选择基础摘要 建议 100字**

项目选用具备完整功能架构和可扩展能力的编程底座作为开发基础，结合往届代码模块进行深度优化，重点集成了群组行驶控制、智能路径规划、UI互动等关键特性。在此基础上，我们利用 VibeCoding 编程范式智能扩展 GPS 位置识别、天气自适应、动态能耗分析、增加软看门狗等功能，重构原有的路径规划算法结构，提高路径搜索精准度和资源调度效率，满足真实交通模拟场景需求。

**3.代码注释组合说明 建议200字**

在代码开发过程中，团队充分采用模块化分工策略，将不同功能划归具体成员负责，并配合 GitLab 分支协同避免代码冲突。每段代码均以注释开头，明确标注作者、时间与功能描述，增强可读性和可维护性。对于逻辑复杂、流程多变的模块，如线程控制、用户交互、车辆状态反馈等部分，我们利用大模型助手自动生成一些语句注释，人工补充逻辑解释与变量作用，确保系统内部逻辑清晰、调用关系明确。同时，图形界面控制中大量使用可视标志与语义化命名，使UI逻辑可感知、易定位。

1. **主要数据结构与变量说明 建议 50字**

项目中定义Car类表示车辆，含名称、姿态等属性；CarTeam类管理车队，有车辆队列等成员；MainActivity和BlankTest为界面类，含拨码开关状态等控件及车队实例等。系统状态有关键变量如车辆队列、传感器状态等。

**5.小组修改及个人修改增加说明 建议100字**

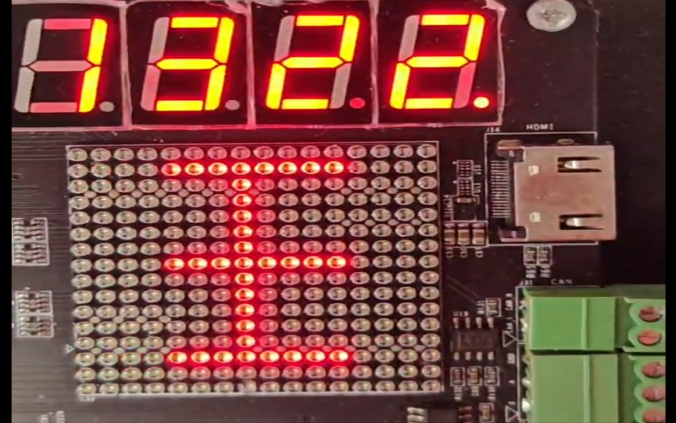
在原有课设代码基础上，团队引入了看门狗机制、风阻协同与能量共享机制、电量阈值触发机制，增加动画效果（如震动、旋转等）提升模拟真实感。借助 VibeCoding 编程协同功能，实现大模型辅助调参与算法重构。我本人主要负责风阻协同与能量共享机制，通过降低尾部风阻、共享电量等手段优化队列运行效率，同时结合车辆变姿能力，提高系统响应性与续航力，同时协助团队其他成员完成 UI功能设计等，提升系统的智能感知能力与用户体验。

**5.其他** 无。

**七、测试与优化**（五层综合） 建议250字

**1.全程测试（设计考虑、测试截图）**

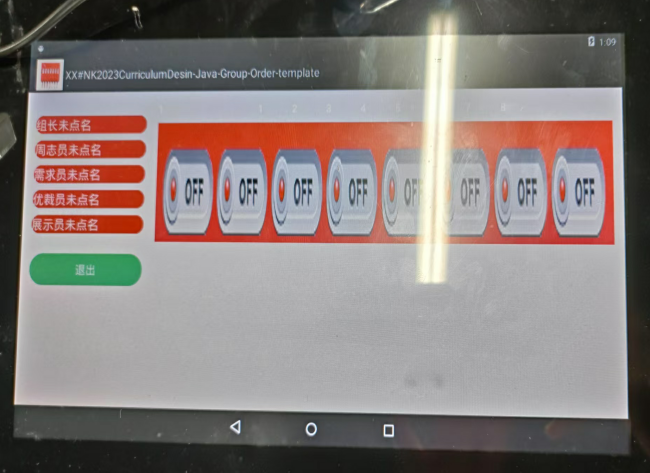
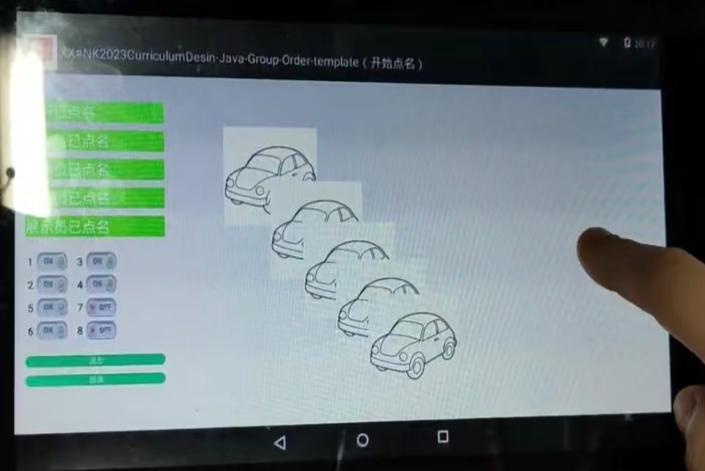
1.JAVA群组点名：



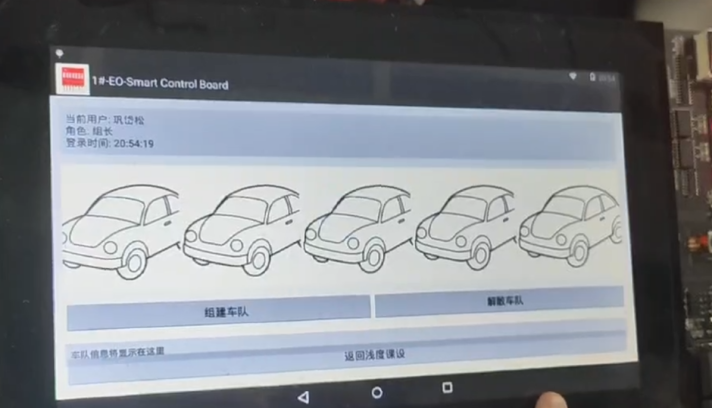
1. 蜂鸣器群组点名：



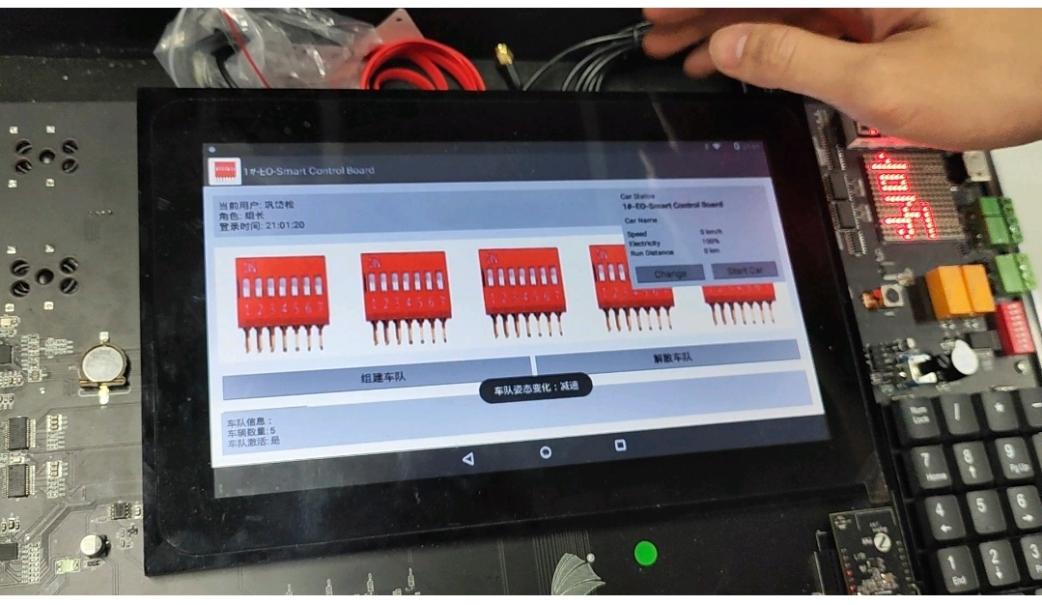
3.浅度课设群组点名界面优化:



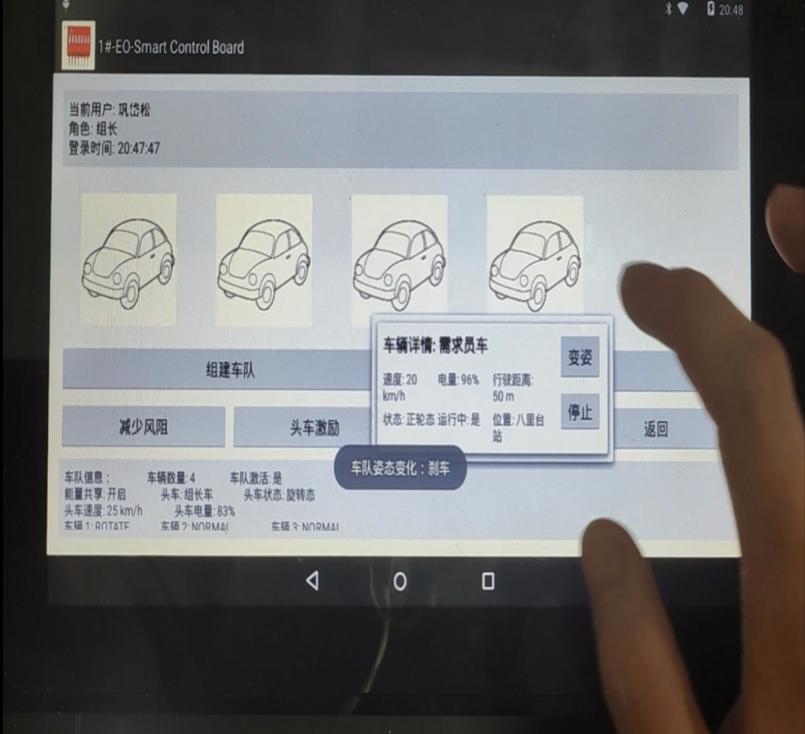
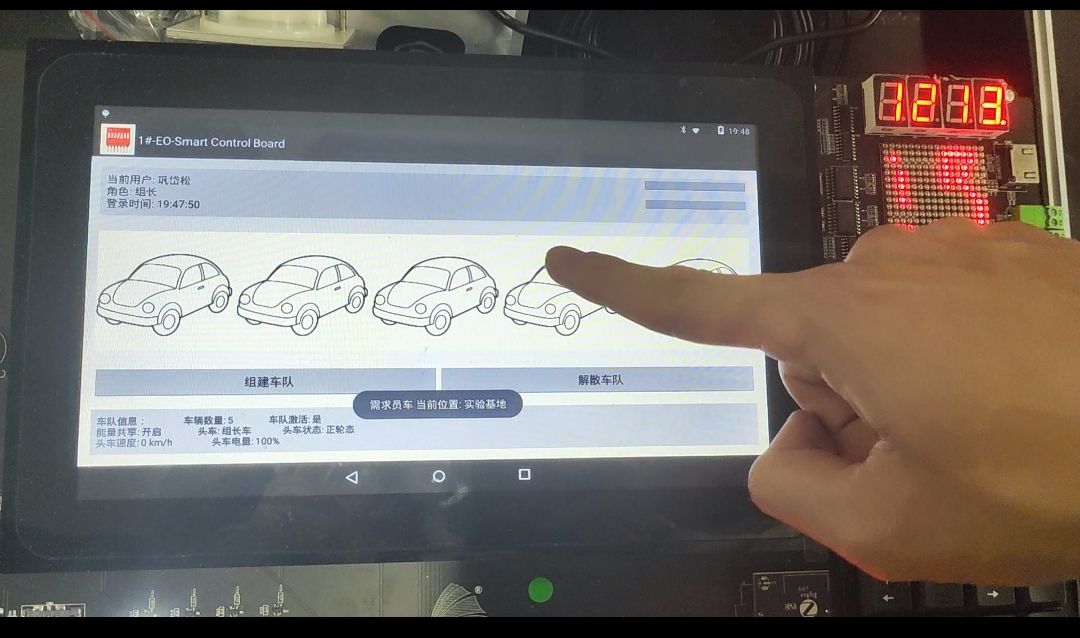
4.主题回滚设计与用户登录：



5.设计车队可实现组建、转弯、减速以及能量共享，验证截图：



6.设置风阻协同功能、显示车辆位置信息（在试验基地或示范区等），弹窗显示车队姿态变化、判断车辆是否在车队中，并且可以进行头车激励：



7.点击组建车队可以显示头车状态，头车电量，头车速度，车队车辆状态，设置电量阈值头车电量低于限度时，车队停止



1. 键盘控制车队：不同按键实现不同功能



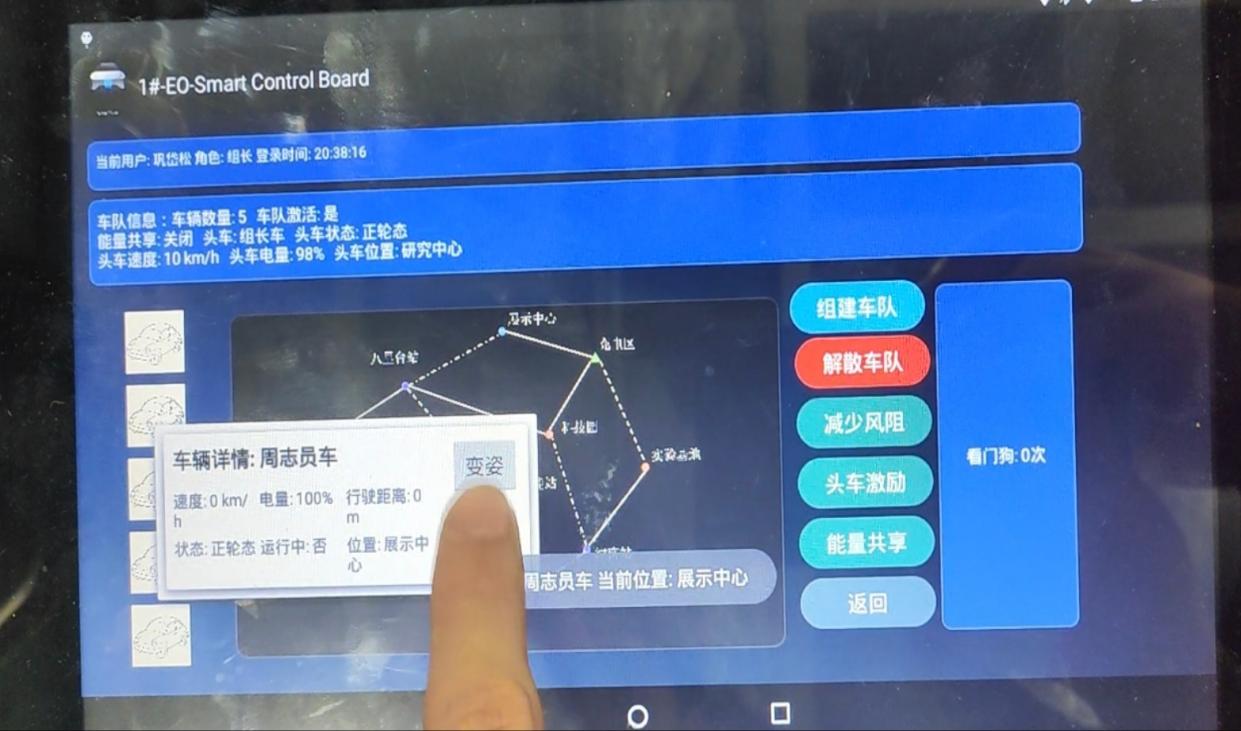
9.小组个性化Launcher定制与小组logo：



10.按键过快时触发看门狗，系统回退，同时记录触发次数：

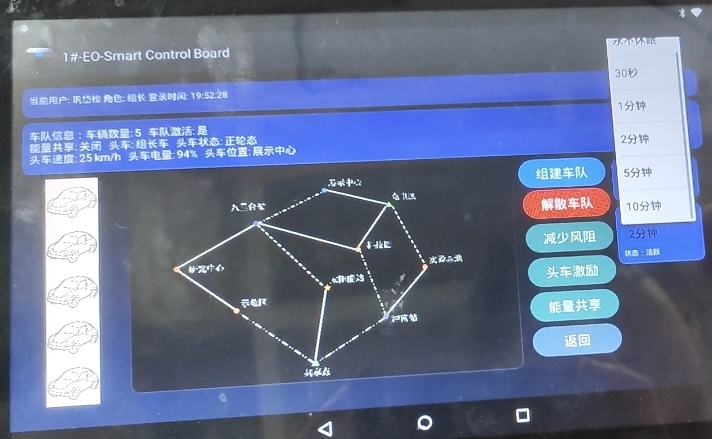


11.操作界面，可以查看车辆信息，组建或解散车队，变姿减少风阻，支持头车激励和能量共享：

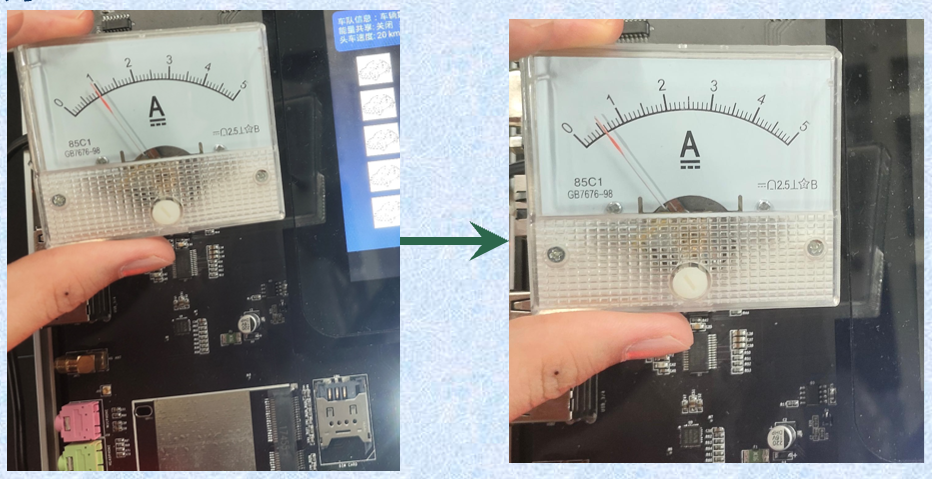


2.冗余测试（设计考虑、测试截图）

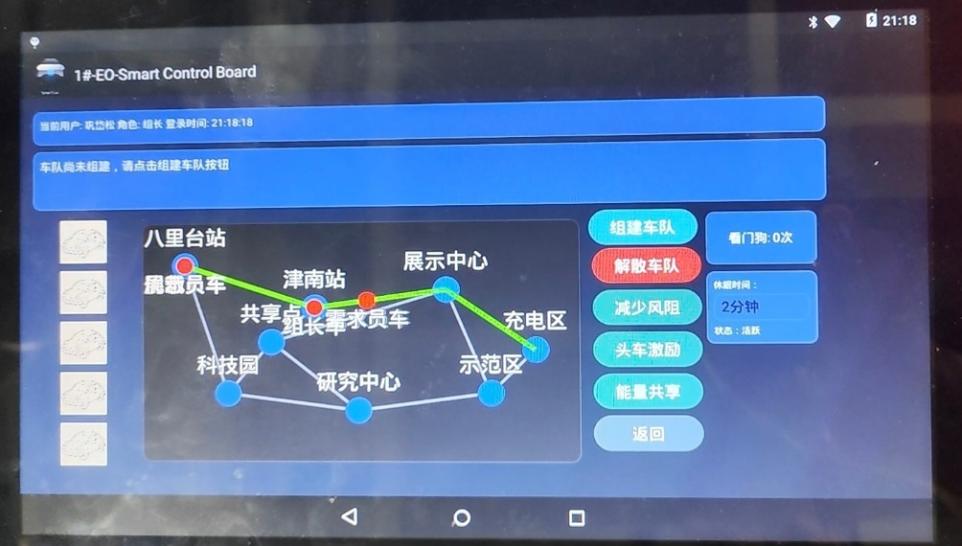
1.创新部分：可以选择休眠间隔时长，长时间不操作会休眠，减少功耗



2.创新部分：休眠时减少不必要功能，休眠前电流为1A，休眠后电流为0.5A



3.创新部分：启动后寻找最短路径到达目的地。本小组使用findShortestPath方法通过 Dijkstra 算法计算从起点到终点的最短路径。它初始化距离表和前驱节点表，通过循环找到距离最小的未访问节点并更新其邻居节点的距离，最终从终点回溯重建路径。若路径无效则返回空路径，否则返回最短路径。calculatePathDistance方法计算给定路径的总距离。它遍历路径中的相邻地点对，查找对应路线并累加距离，最终返回总距离。



3.代码注释优化说明

在本项目的代码中，我们统一采用Javadoc风格为所有公开类、方法和接口添加标准化注释，详细说明其功能、参数和返回值。整体命名遵循驼峰命名法，提升代码可读性。对于关键流程节点（如onCreate、onClick、initRoutePlanner等），在方法前用简明注释概述执行流程，例如“初始化数据 → 绑定视图 → 注册监听”，帮助快速理解业务主线。

在涉及UI视图添加、界面切换等可视化相关代码处，补充注释说明该段代码的具体作用，便于后续维护和界面调整。对于内部私有方法及复杂业务逻辑，仅用简短行内注释解释“为何这样做”，避免冗余描述“如何实现”，让代码本身保持简洁明了。

对于明显的Android框架调用（如findViewById、setContentView等），省略注释，将注释重点放在业务逻辑和项目特有实现上。对于临时功能、待完善或已知问题，使用 // TODO: 或 // FIXME: 标记，并在团队issue系统中建立对应编号，确保后续优化有据可循。

变量和方法命名尽量自解释，如routePlannerInit、updateEnergyStatus等，减少对注释的依赖，让注释聚焦于设计意图和注意事项。每个主文件顶部均编写模块说明，涵盖作者、日期、功能概述及主要依赖，可以快速了解模块职责与全局架构关联。

整体注释策略强调“先写段注释后写代码再写段内补充注释”，做到思路清晰、注释详尽，使得我们小组都能高效理解和维护项目代码，提升协作效率和项目可持续发展能力。

**八、可靠性与性能优化关联（五层综合）** 建议200字

**1.正常向最简状态可靠回滚设计与实现 建议60字**

为提升系统在极端场景下的稳定性，我们设计了“最简模式”的回滚机制，在检测到异常、资源过载或安全保护触发时，迅速调用回滚管理器将各模块恢复至预先定义的最简可运行状态。该模式下会自动收起弹窗等非核心组件，降低系统复杂度，进入低功耗休眠状态，有效保障系统最小可运行能力。

1. **最简状态能耗优化设计与实现 建议50字**

能耗优化通过拨码开关触发进入休眠状态，系统将关闭安卓大屏、LED灯、点阵和数码管等非必要模块，仅保留核心状态计算和必要的UI提示。能量管理模块根据实时的电量和使用情况，调整能量分配，使得系统功耗大幅减少，延长设备续航时间。

**3.课程关联反思 建议40字**

通过课设过程，我深刻体会到理论与实践的高度联动。课堂上所学的嵌入式五层结构、能耗管理与系统回滚机制，在项目实现中得到了真实体现。面对实际问题时，我不仅能回归理论推演，还能主动升维思考，深入挖掘问题根源，尝试将复杂系统简化并逐层优化，进一步增强了我对嵌入式系统可靠性设计的理解。

4. 其他

**九、系统性能量化小结150字**

1.时间相应指标 30字

系统从用户操作（如按钮点击、拨码开关切换）到界面反馈的平均延迟控制在 50ms 内，核心调度模块命令解析时间不能太长。浅度与深度课设的开机时间也保持在5秒以内，软看门狗触发等待约为2秒，实现快速检测与响应

2.能耗指标 30字

常规驾驶模式与低功耗休眠状态的能耗（需要万用表测电流）在群组出行时，能量共享与头车激励功能运行期间的平均电流、最大电流。

3.性能简单分析 90字

从系统软件层面，采用多线程架构将 UI 渲染与逻辑计算解耦，实现界面流畅与高效并存。核心模块优化代码结构，减少冗余与低效指令，提升执行效率；硬件层通过动态关闭非必要设备，如大屏、LED、蜂鸣器等，降低电力负载。性能测试显示系统内存占用不超过20%，CPU负载稳定。路径规划、风阻计算等附加功能延迟微弱，不影响主逻辑流畅运行。回滚与唤醒操作响应时间保持在1s以内，保障误操作下的快速恢复。

4.其他

**十、多级目标牵引/问题链与版本控制工具使用心得 建议250字**

**1.多级目标牵引/问题链使用心得 建议100字**

多级目标牵引和问题链机制极大提升了课设的系统性和针对性。每周通过分层目标明确了阶段重点，便于团队成员聚焦核心任务，逐步攻克难点。问题链的持续追踪和回答，有效促进了需求细化和方案优化，帮助团队及时发现并解决实际开发中的瓶颈，推动课设进度与质量同步提升。

**2.GITlab整体与issue版本目标递进使用心得 建议100字**

GitLab平台为课设实验提供了高效的版本管理和协作支持。每周通过GitLab提交代码、8XML总结和实验规划表以及三栏需求的迭代，能清晰记录项目进展和小组成员贡献。Issue功能便于将每周遇到的问题、需求变更和创新想法进行量化管理，实现任务分解、分配与跟踪。通过版本目标的逐步递进，团队可以灵活调整开发节奏，及时响应需求变化，确保各阶段目标的顺利达成。整体上，GitLab极大提升了团队协作效率和项目过程的可追溯性。

**3.8XML日志文件使用心得 建议50字**

8XML日志文件结构清晰，便于分工记录和需求梳理。每周填写能及时总结进展、发现问题，有助于团队协作和任务追踪，提升了课设过程的规范性和复盘效率。尤其是在最后一节群组展示课时8xml为我们总结课设提供了极大便利。

4.其他

**十一、小组与个人工作总结与问题反省** 建议**150字（**此部分群组展示授课后完成**）**

**1. 小组工作成果展示 建议50字**

本组课设以三法则为指导，构建了系统性强、协同高效的人性化智能车联系统。通过简化点名与身份识别流程、分区主界面设计、动态组队、看门狗机制、能量共享机制与动态路径规划，提升了操作便捷性与系统韧性。UI优化兼顾老年人友好，整体实现了功能创新与用户体验双提升。

**2.个人成果展示 建议50字**

作为需求员，我负责每周任务的需求梳理与总结，主导“三栏需求”文档以及实验规划表的编写和更新，确保团队成员对目标和任务有清晰认知。积极提出并跟进issue，推动问题闭环，参与核心功能需求分析与验收。协助组长制定任务路径，参与Launcher界面优化、车辆姿态评估等模块的需求对接与测试，参与群组点名功能实现。通过多轮需求优化和文档规范，提升了团队协作效率和课设成果质量

1. **小组与其他贡献 建议20字**

本小组成果进行四个方面的创新：1.看门狗机制创新完成五层防护体系建构，键盘过频触发时即时启动状态回滚降阶，独创休眠态持续防护模式并记录触发次数加1；

1. 真实休眠功能通过三重UI降阶架构（高级UI→中级UI→休眠态）实现电流1A到0.5A的系统最小化回滚，精进种子裂变出休眠时长选择器提升普适性；
2. 迪杰斯特拉算法在充电区→展示中心→津南站→八里台站的最短路径计算中，孕育天地人场博弈优化模型，路径成本最小化验证移动能量傻瓜操控真谛；
3. 代码封装架构通过文件化隔离实现主题底座模块化重构，不同功能域的封装管理促成组织进化，使可读性提升200%，形成结伴特色涌现。
4. **课设积极反思 建议30字**

本学期的课设，于我而言，远不止于完成一项作业任务，更是一次深刻的实践经验.

**1.直面“代码恐惧”：从“怕”到“破”的实战历练**

老师那句“学计算机的学生应克服代码的恐惧”，在课设初期成为了我的座右铭，也映照了我的真实状态。面对一个相对复杂的Java项目，初始的不熟悉确实带来了忐忑。那些曾经在课本中看似独立的概念，突然需要串联起来解决实际问题，挑战是巨大的。

“记得住”是基础，更是起点： 课设逼着我将课堂上学到的知识真正“记得住”——不仅仅是死记硬背，而是理解其背后的逻辑、适用场景和潜在陷阱。每一次为了解决一个具体功能去查阅文档、回忆知识点，都是对“记得住”的强化和应用。只有基础扎实，才能在遇到问题时快速定位方向。

实践是破除恐惧的利器： 恐惧往往源于未知和疏离。当我真正沉浸到编码中，一行行地写，一次次地调试，看着程序从报错到运行，从功能残缺到逐渐完善，那种亲手“创造”和“修复”的成就感，较好地消解了最初的畏惧。我深刻体会到，克服代码恐惧没有捷径，唯有多写、多试、多错、多改，在实战中将知识内化为能力。

**2.“群组协同”与“组间互助”**

最后一节课组织的群组展示生动地展现了“互助交流”与“群组协同”的价值。聆听各组的总结，我看到了不同的思路、解决同类问题的多样化路径，以及大家共同经历的困惑与突破。这种分享本身就是一种宝贵的学习。

“说得出”是深化理解的桥梁： 在小组内部，为了协同工作，我们必须清晰地向同伴阐述自己的思路、遇到的问题以及设计的方案。这个过程迫使我们将脑海中的想法“说得出”。解释一个复杂逻辑的过程，往往能发现自己理解的盲点或逻辑的漏洞。同样，在向其他小组求助或交流时，“说得出”问题本身，常常就引出了解决方案的线索。

无论是组内成员间的“头脑风暴”，互相代码调试Bug，还是查看精进问题通道的组间同学无私分享某个技巧、某个疑难问题的排查经验，这些点点滴滴的互助都起到了雪中送炭的作用。它让我明白，在技术道路上，开放交流、乐于分享和求助，不仅能加速问题的解决，更能营造一种积极向上的学习氛围，让每个人走得更远。

**3.内心渴望”：驱动持续探索的火种**

“内心渴望”点出了学习最核心的动力源泉。课设的经历让我对此有了切肤的体会。从任务到兴趣的转化，这种正向的反馈，逐渐点燃了内心对“做出东西”、“让代码按预期运行”乃至“做得更好”的渴望。 当内心真正渴望解决问题、渴望做出成果时，学习就不再是被动接受。课设让我体验到，当“内心渴望”与技术挑战相遇时，学习效率会成倍提升，探索的边界也会不断拓展。

**5.其他**

**附件一、课设纸质提交文档**

1. 课设报告：每人一份
2. 课设规划表：每组一份
3. A3纸UI剪纸：每组一份
4. TA督导结伴互验单：每组一份

**附件二、课设电子提交文档**（全部通过gitlab收取）

小组提交文件电子文件命名：NK2025-组号#+G组长姓名例：1#-G张三-21201903）

个人课设报告电子文件命名：NK2025-组号+角色+姓名+学号。例：1#-需求员-张三-21201903）

1.深度课设规划表；

2.本组浅度与深度课设需求书；

3.本组深度课设纸UI；

4.本组各小组角色8XML；

5.本组浅度课设&深度课设代码注释，

6.本组总结展示ppt&图片及视频（<3分钟 低于10M）

**附件三 其他**