

基于 RK3588 的车载辅助驾驶终端

摘要

为有效化解驾驶途中潜藏的各类安全隐患，同步大幅提升驾驶的舒适度与便捷性，本项目依托性能强劲的 RK3588 芯片，精心研发了一款功能完备的车载辅助驾驶终端。该终端采用先进的 YOLO11n 模型，并结合 OpenCV 视觉处理库，能够精准追踪道路上的行人与过往车辆。系统针对车载环境特点对 YOLO11n 进行了模型压缩与量化优化，在保持高检测准确率的同时，显著提升推理速度，满足系统对实时性的要求。当监测到行人靠近车辆一定范围时，系统会立即触发警告机制，通过时空关联分析预测潜在风险，并向驾驶者发出预警。

在视觉呈现方面，借助 OpenGL 技术对检测到的车辆与行人进行三维渲染建模。通过优化渲染管线，实现检测数据的实时可视化，将行人、车辆等信息以直观清晰的方式呈现到终端显示屏上，增强驾驶者对周边环境的感知能力。

同时，终端集成了 LD2451 毫米波雷达模块，用于监测车辆后方区域。该雷达能够实时检测后方来车、障碍物等目标，通过计算相对距离和速度，判断是否存在安全隐患。当车尾安全空间不足时，系统会根据风险等级发出相应的预警信号，为车辆后方安全提供保障。

此外，搭载的 BH1750 光线传感器能实时感知外界光照强度，系统基于环境光照数据，自动控制车辆灯光的开启与关闭，在不同光照条件下均能确保良好的驾驶视野，同时避免手动操作灯光带来的分心。

在多维度技术的融合应用作用下，该车载辅助驾驶终端可实现对路况的精准感知与智能分析，为驾驶者提供实时、全面且清晰的周边环境动态，助力驾驶者在复杂路况下快速做出正确决策，从而有效降低交通事故发生率，显著提升行车安全性与驾驶体验。

第一部分 作品概述

1.1 功能与特性

该项目基于 RK3588 芯片实现了以下关键功能及特性：

(1) **目标检测与追踪**：集成 YOLO 系列算法，实时检测并跟踪前方车辆与行人，实现复杂场景下的高精度识别。

(2) **行人靠近预警**：通过图像识别与空间计算判断行人接近程度，并进行声光提示。

(3) **三维可视化**：利用 Blender 建模与 OpenGL 实时渲染，将车辆、行人、车道等检测结果以 3D 形式直观呈现，提升信息清晰度。

(4) **后方来车雷达预警**：通过 LD2451 毫米波雷达感知后方来车，实时发出靠近警告，提升变道与倒车安全。

(5) **环境光自适应灯控**：结合 BH1750 光线传感器，在夜晚或光线较弱场景下自动开启车灯，增强可视性。

系统融合视觉感知、雷达感知与环境感知技术，为驾驶者提供全面、精准的辅助决策支持。

1.2 应用领域

本车载辅助驾驶终端凭借精准的环境感知、实时的风险预警及高效的场景适配能力，可广泛应用于多元交通场景，为不同出行需求提供安全与体验升级支持：

(1) **个人私家车**：随着交通流量持续增长，交通事故风险随之攀升，私家车主对主动安全防护的需求愈发迫切。本系统通过实时监测车辆周边行人、车辆动态及车道线状态，为私人驾驶场景构建全方位安全防线，有效降低剐蹭、碰撞等事故发生率，同时以直观的信息呈现减轻驾驶压力，提升日常出行的舒适与安心感。

(2) **共享出行**：在共享汽车、网约车等高频使用场景中，车辆安全性能直接影响用户体验与平台口碑。系统搭载的行人检测、车道线识别及碰撞预警功能，能为每一次行程提供标准化安全保障，减少因驾驶员经验差异导致的风险，助力平台提升服务质量与用户信任度。

(3) **物流和运输**：货运与快递行业中，长途驾驶、复杂路况及时效压力易引发安全隐患，且事故造成的损失往往较大。本系统通过精准感知周边环境，辅助驾驶员预判潜在风险，优化车道保持与跟车距离，在保障运输安全的同时，减

少因事故导致的延误，间接提升运输效率与经济效益。

(4) 公共交通：公共汽车、出租车等承担着城市大规模人员运输任务，其安全性关乎公众出行信心。集成本系统后，可通过实时监测行人横穿、车辆加塞等突发情况，为驾驶员提供及时提醒，降低恶性事故发生概率，增强公众对公共交通的安全感与依赖度。

(5) 智能城市建设：在智能交通体系加速构建的背景下，本系统可作为移动感知节点，与交通信号灯、路侧雷达等基础设施联动，实时反馈路面动态数据，为城市交通流量调控、事故快速响应提供支撑，助力打造更安全、高效的智慧交通生态。

(6) 驾驶培训：驾驶学员因经验不足，常对周边环境变化反应迟缓，易发生操作失误。本系统能实时标注行人、车辆及车道边界，辅助学员建立环境感知意识，规范驾驶操作习惯，缩短培训周期的同时，降低练习过程中的事故风险。

(7) 老年人独立出行：老年驾驶者受生理机能影响，反应速度与视野范围可能受限，独立出行存在一定安全顾虑。系统提供的实时预警、辅助判断功能，可弥补其感知短板，帮助老年人在日常出行中更从容应对复杂路况，保障独立出行的安全性与自主性。

1.3 主要技术特点

(1) YOLO 目标检测：采用 YOLO 系列算法进行实时目标检测与追踪，能够高效识别和定位周围的车辆和行人，为驾驶者提供精准的周边环境信息。该算法以其高效性和准确性，确保低延迟的实时反馈。

(2) RKNN 格式模型导出：通过将 Ultralytics YOLO11 模型转换为 RKNN 格式，系统能够在 Rockchip 硬件平台上充分利用其内置的神经处理单元(NPU)，实现更高的推理效率。RKNN 模型优化了在 NPU 上运行的性能，减少了推理延迟，同时降低功耗，有效延长了便携设备的电池寿命。这种针对 NPU 的优化确保了在执行人工智能任务时的快速响应与高能效，使得嵌入式 AI 应用能够在边缘设备上流畅运行。

(3) 三维绘制技术：利用 OpenGL 进行三维模型的实时渲染，支持对车辆

和行人的动态可视化，确保驾驶者能够实时观察到周围的交通情况。

（4）雷达集成：集成 HLK-LD2451 短距离雷达模块，实现后方来车警告。雷达能够快速和准确地检测到路面上的其他车辆，提供实时距离和角度信息，进一步增强安全性。

（5）交互式界面设计：基于 Pygame 构建的用户界面可提供友好的交互体验，驾驶者可以通过简单操作访问不同的功能模块，包括道路智能识别和雷达设置等，提高用户的便捷性。

（6）环境感知功能：综合使用 YOLO 目标检测和雷达数据，系统能智能判断周围环境并在必要时实时发出安全警告，增强行车时的安全意识，降低潜在事故风险。

（7）多模式雷达配置：用户可根据不同驾驶场景选择雷达工作模式（如普通道路、高速道路、倒车预警等），确保在各类场景下最大限度地保护行车安全。

（8）实时数据更新：通过对雷达和相机获取的数据进行整合与分析，系统可动态更新周边环境信息和目标位置，给予驾驶者明确的距离信息，助力快速决策。

1.4 主要性能指标

本项目主要性能指标如下：

- 车辆/行人检测准确率（F1 SCORE）：98.5%
- 视频流采集成功率：100%
- 雷达数据采集成功率：99%
- AI 推理平均延迟：30ms/帧
- 3D 渲染与 UI 显示帧率：60FPS~90FPS
- 系统稳定运行时长：≥72 小时
- 打包后可执行文件兼容性：100%（RK3588 ARM64 平台）
- 多模型/多传感器扩展成功率：100%
- 信息传输成功率：100%

1.5 主要创新点

(1) RKNN 模型优化：将 Ultralytics YOLO11 模型转换为 RKNN 格式，并在 Rockchip NPU 上运行，实现高效的目标检测与推理，显著提升了处理速度与响应时间。项目所需 RKNN 文件如图 1 所示。

metadata.yaml	2025/07/09 13:51	YAML 文件	2 KB
yolo11n-rk3588.rknn	2025/07/09 13:51	RKNN 文件	7,550 KB

图 1 RKNN 文件

(2) 雷达集成：集成 LD2451 短距离雷达提供精准的后方监测和警告功能，增强了行车安全性，帮助驾驶者识别潜在风险。

(3) 多模式雷达配置：允许用户根据不同驾驶场景（如普通道路、高速公路、倒车）选择雷达工作模式，提高系统的灵活性和适应性。多模式雷达运行实图如图 2 所示。

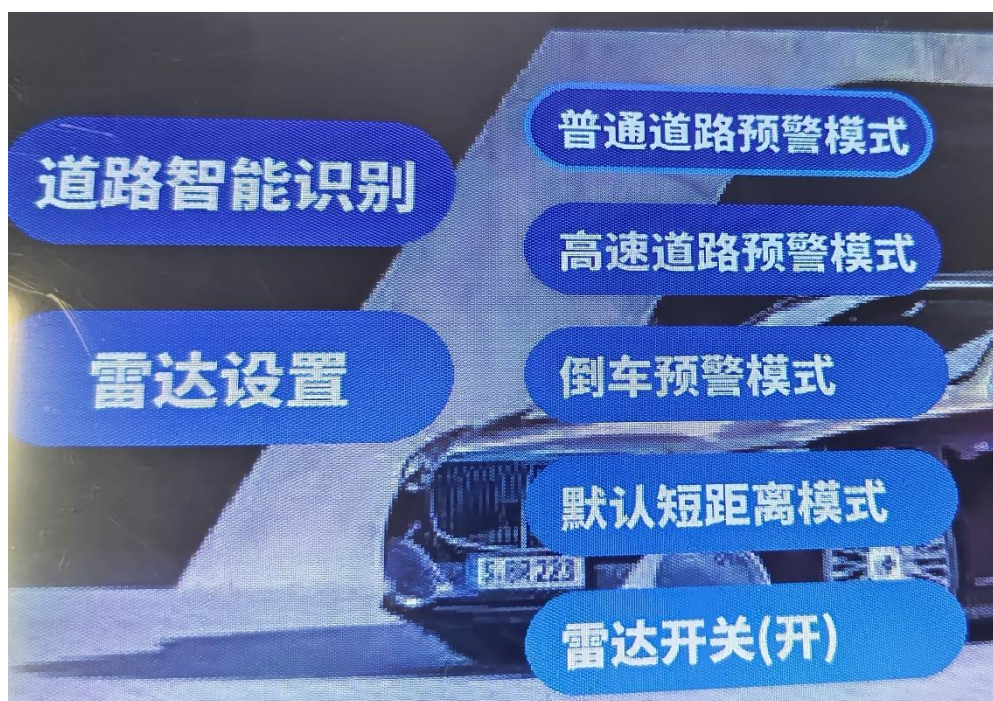


图 2 多模式雷达运行实际图

(4) 用户友好界面：基于 Pygame 构建的交互式用户界面使功能访问直观便捷，提升用户体验，确保驾驶者能快速获取关键信息。

(5) 深度学习与传感器融合：通过结合深度学习技术与雷达数据，实现智

能环境感知，实时发出安全警告，减少事故风险。

1.6 设计流程

（1）系统选型与环境配置

本项目根据 ARM64 架构，选择合适的 Ubuntu 操作系统作为开发平台，以支持高效的图形渲染和模块化开发。

（2）模块化开发与逐步实现

1) 视觉信息处理模块

① 视觉检测功能

利用 YOLO 系列算法实现对行人和车辆的实时检测，通过深度学习模型分析视频流，准确识别和分类周围交通对象。

② 实时数据渲染

基于 OpenGL 库，将检测到的行人、车辆及车道线信息实时渲染于终端界面，实现直观的可视化效果，提升用户体验与交互性。

2) 雷达感知模块

① 雷达功能测试

针对 LD2451 雷达进行实验与验证，以评估其在检测后方来车警告方面的性能与精度。

② 雷达子功能设计

设计并实现多个雷达应用功能，以支持多种驾驶环境下的安全预警与辅助决策。

3) 功能集成与系统优化

将上述各模块进行系统集成，确保各功能模块之间的高效协作与信息共享，最终优化整体系统的性能与稳定性，确保实现精确的路况感知和安全的驾驶辅助体验。本项目的设计流程示意图如图 3 所示。

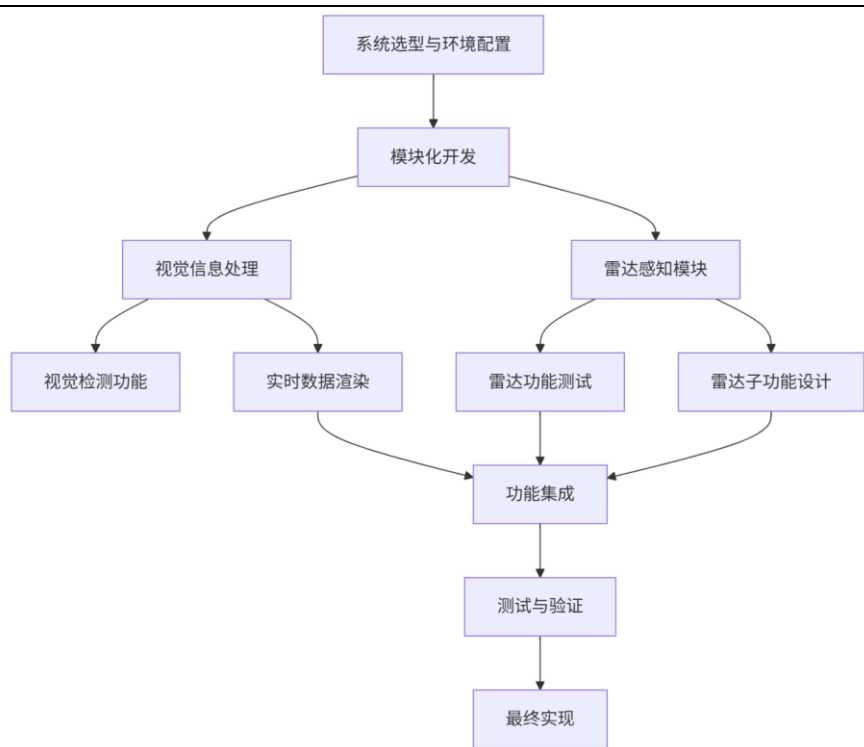


图 3 设计流程示意图

第二部分 系统组成及功能说明

2.1 整体介绍

该项目的系统整体框图如图 4 所示。

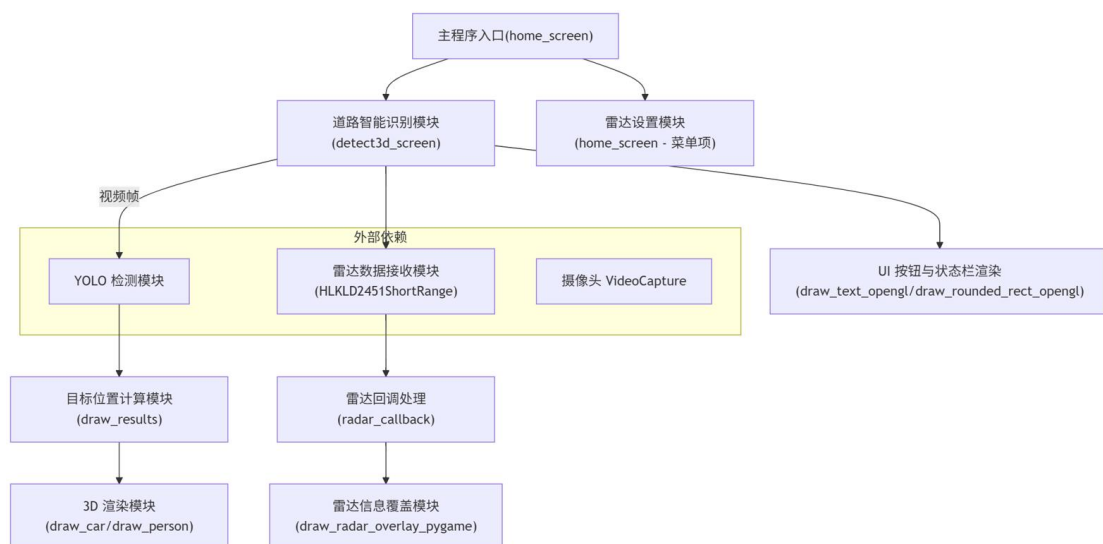


图 4 系统整体框图

(1) 主程序入口 (home_screen)

启动后进入全屏主界面，提供“道路智能识别”与“雷达设置”两个主要按钮。

用户可在“雷达设置”中开启/关闭雷达或切换预警模式，配置结果保存在全局 `radar_setting`。

（2）道路智能识别模块 (`detect3d_screen`)

由主界面点击“道路智能识别”进入。

初始化 OpenGL 透视投影、摄像头采集、雷达连接。

在循环中同时处理视频帧和雷达数据，并渲染 3D 场景与 UI。

（3）YOLO 检测模块

对每帧输入图像执行目标检测，输出包含车辆和行人的边界框与置信度。

（4）目标位置计算模块 (`draw_results`)

根据检测框底部坐标，通过相机模型（焦距、相机高度）计算实际距离与度。

分别组装车辆和行人位置列表，并去重/防重叠。

（5）3D 渲染模块 (`draw_car / draw_person`)

接收由位置计算得到的坐标，在 OpenGL 三维场景中绘制简易模型。

车辆采用红色与灰色面，行人采用蓝色模型。

（6）雷达数据接收模块 (`HLKLD2451ShortRange`)

封装串口连接、模式配置、数据接收线程。

根据 `radar_setting.mode` 发送相应配置命令。

（7）雷达回调处理 (`radar_callback`)

将解析后的多目标数据选取最近一个，更新全局 `latest_distance` 与 `latest_angle`。

（8）雷达信息覆盖模块 (`draw_radar_overlay_pygame`)

在 OpenGL 渲染之上，使用 Pygame 字体将最近距离/角度以像素方式叠加绘制。

（9）UI 按钮与状态栏渲染(`draw_rounded_rect_opengl`)

在 3D 场景顶部和右侧绘制返回按钮、状态条、行人警告框等，接收鼠标点击进行界面跳转或退出。

2.2 硬件系统介绍

2.2.1 硬件整体介绍

采用 HLK-LD2451 雷达模块、BH1750 光照模块、OV13855 摄像头和 RK3588 组成的智能车载终端，其实物图如图 5 所示。

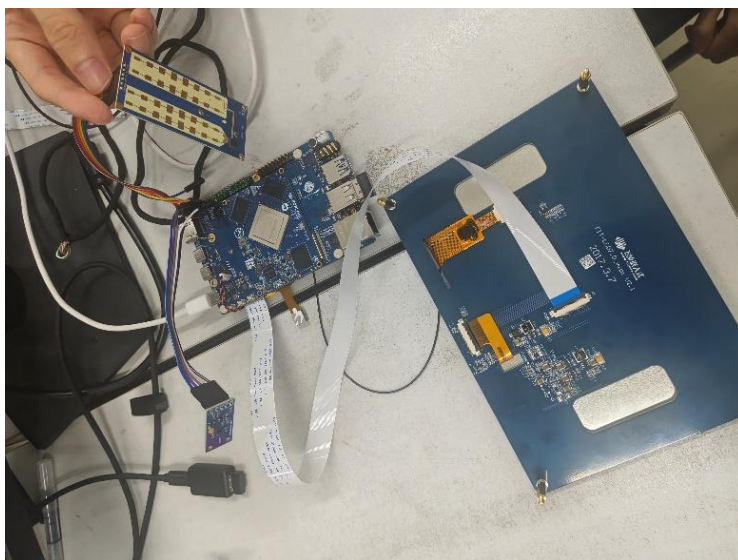


图 5 智能车载终端实物图

2.2.2 电路各模块介绍

(1) 智能车载终端整体

CSI2 DPHY: OV13855 摄像头通过 4 路 MIPI-CSI2 与 RK3588 相连，用于实时视频采集。

I²C: BH1750 光照模块挂在 I²C 总线上，可用于自动大灯控制。

UART2: HLK-LD2451 雷达模块通过串口与 RK3588 通信，实时距离与角度数据的传输。

HDMI/eDP: 输出给车载屏，用于显示三维渲染画面和 UI。

车载终端模块划分图如图 6 所示。

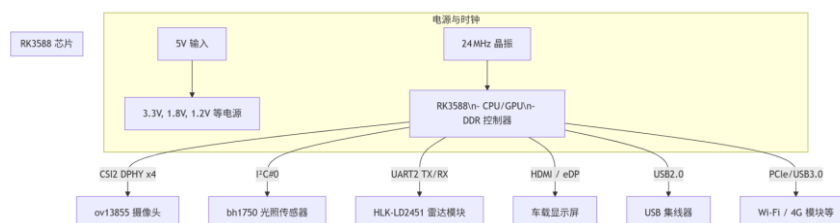


图 6 模块划分图

(2) HLK-LD2451 雷达模块

感知区域内是否有靠近或远离的车辆，实时输出检测结果。最远感应距离可达 100m。检测后方视野盲区，避免后方有车辆靠近突然变道发生车祸和倒车障碍物检测。雷达模块实物连接如图 7 所示。

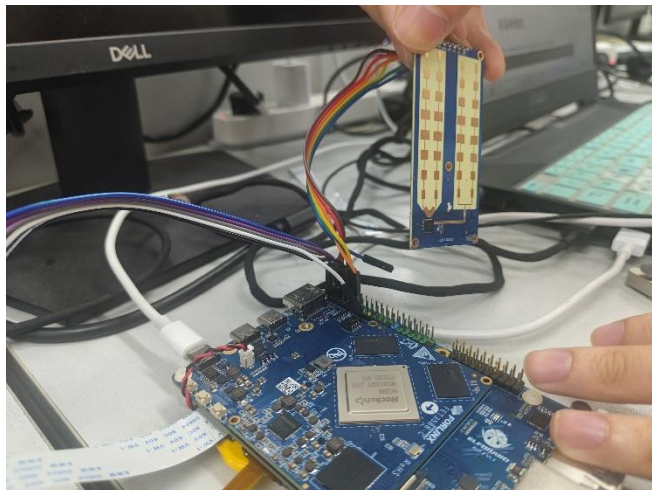


图 7 雷达实物接线图

串口需做 5V→3.3V 的电平转换，以保护 RK3588 的串口。

模块工作电流峰值约 100 mA，需在电源插座处加去耦电容。HLK-LD2451 雷达模块硬件图如图 8 所示。

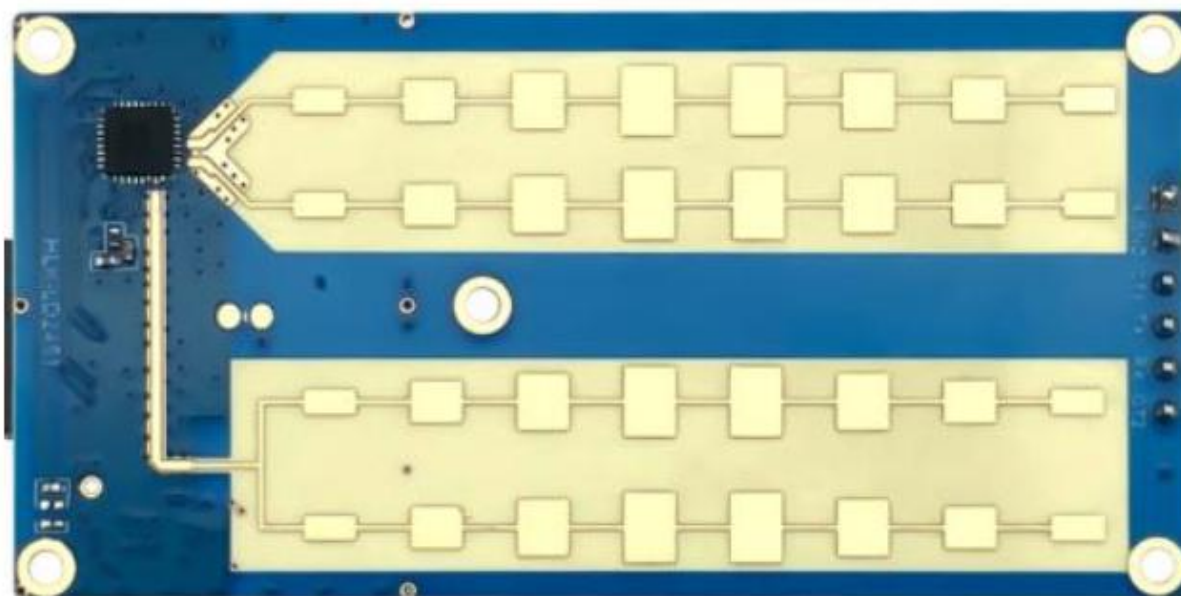


图 8 HLK-LD2451 雷达模块硬件图

(3) OV13855 摄像头

用于视觉扫描，OV13855 摄像头作为一款高性能图像采集设备，凭借其出色的成像能力，成为本车载辅助驾驶终端视觉扫描模块的核心组件。该摄像头具备高分辨率成像特性，能清晰捕捉车辆周边的行人、车辆及车道线等关键信息，为后续的 YOLO11n 目标检测算法提供高质量的原始图像数据。视觉扫描模块如图 9 所示。

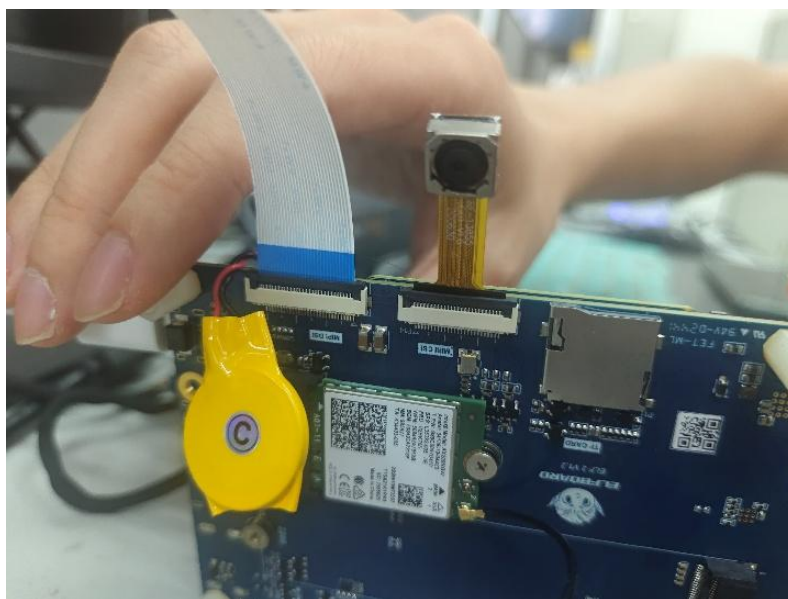


图 9 视觉扫描模块

(4) BH1750 光照模块

BH1750 光照模块作为系统环境感知的重要组成部分，凭借其高精度的光强检测能力，为车载辅助驾驶终端的自动灯光控制功能提供关键数据支撑。具备快速响应特性，可实时捕捉外界光照强度的细微变化。车载终端的光照模块如图 10 所示。

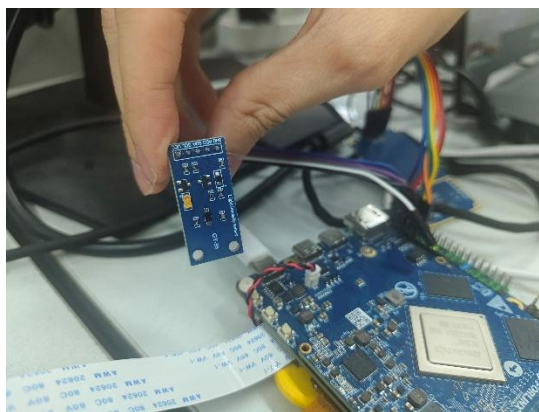


图 10 光照模块

2.3 软件系统介绍

2.3.1 软件整体介绍

本智能车载终端软件采用模块化设计，主要包括摄像头与雷达数据采集、AI 推理、3D 渲染与 UI 显示、数据管理等功能模块。软件支持多路视频流实时处理，集成了车辆与行人检测、距离测量等 AI 算法，并可根据需求切换不同模型和推理后端。UI 界面友好，支持警示提示、状态显示等交互功能。系统通过 PyInstaller 打包，便于在 ARM64 平台一键部署，具备良好的兼容性和可扩展性，适用于智能驾驶辅助、车载监控等应用场景。

2.3.2 软件各模块介绍

系统顶层架构主要模块：

1. 主控程序
2. 摄像头采集模块
3. 雷达数据采集模块
4. AI 推理模块（YOLO11n→RKNN）
5. 3D 渲染与 UI 显示模块
6. 数据管理与日志模块

各模块设计说明：

主控程序

功能：负责系统初始化、各模块调度、异常处理和主循环控制。

关键输入：配置文件、命令行参数、外部事件

关键输出：系统状态、日志、UI 刷新信号

流程图如图 11 所示

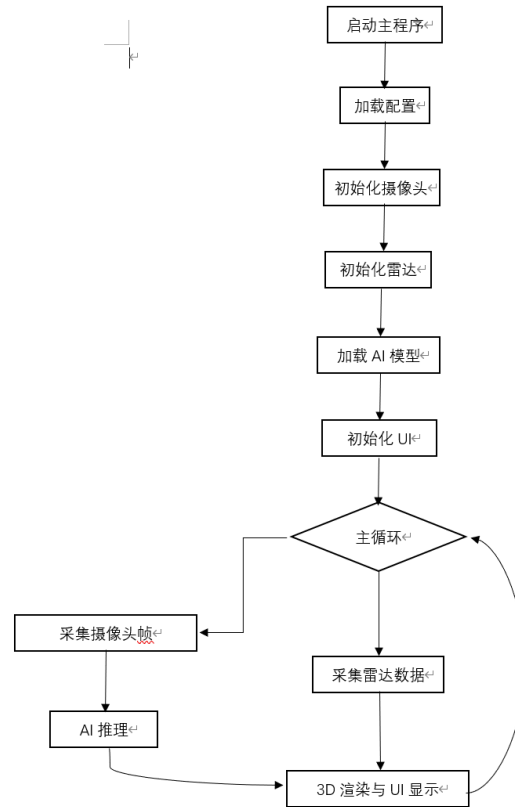


图 11 主控程序流程图

摄像头采集模块：

功能：从/dev/video11 采集图像帧，预处理后送入 AI 模块

关键输入：摄像头设备节点、分辨率参数

关键输出：原始/预处理图像帧

流程图如图 12 所示



图 12 摄像头采集模块流程图

雷达数据采集模块：

功能：通过串口/dev/ttyS9 读取 HLK-LD2451 雷达数据，解析目标信息

关键输入：串口号、波特率、雷达协议

关键输出：目标距离、速度、坐标等

流程图如图 13 所示

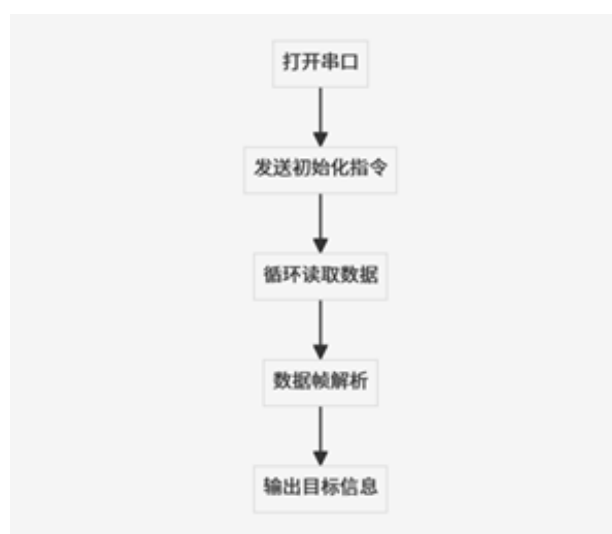


图 13 雷达数据采集模块流程图

AI 推理模块 (YOLO+RKNN):

功能: 加载 RKNN 模型, 调用 rknnlite 进行行人/车辆检测

关键输入: 预处理图像帧、模型文件、类别元数据

关键输出: 检测框、类别、置信度

流程图如图 14 所示



图 14 AI 推理模块流程图

3D 渲染与 UI 显示模块

功能: 基于 OpenGL/pygame 渲染检测结果、雷达点云和 UI 界面

关键输入: 检测结果、雷达目标、系统状态

关键输出: 屏幕显示帧、用户交互事件

流程图如图 15 所示



图 15 3D 渲染与 UI 显示模块流程图

数据管理与日志模块

功能：记录检测、雷达、系统日志，支持本地存储与远程上传

关键输入：检测结果、雷达数据、系统事件

关键输出：日志文件、上传数据包

流程图如图 16 所示

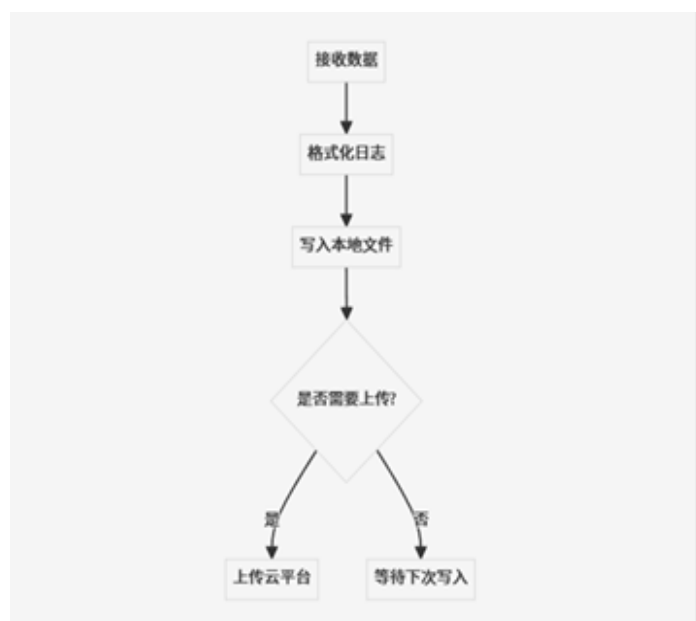


图 16 数据管理与日志模块

第三部分 完成情况及性能参数

3.1 整体介绍

本系统基于 RK3588 开发板，集成摄像头、雷达与 AI 推理，实现车载环境下的行人和车辆检测、3D 可视化显示。系统采用模块化设计，涵盖数据采集、AI 推理、3D 渲染、UI 交互与数据管理，支持多传感器扩展和云端对接。通过软硬件协同优化，保障终端高效、稳定运行，适用于智能车载终端场景。实物整体外观如图 17、18 所示。



图 17 整体实物图 1

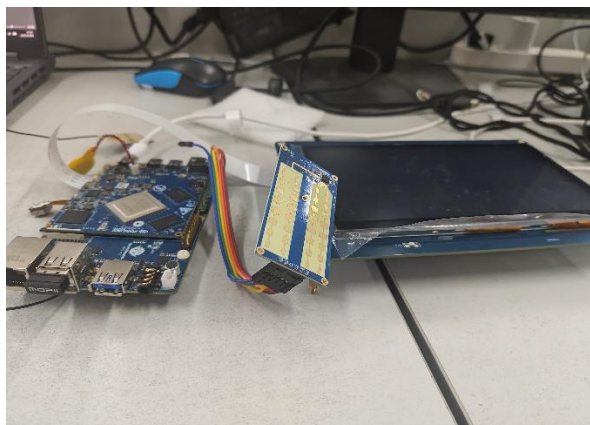


图 18 整体实物图 2

3.2 工程成果（分硬件实物、软件界面等设计结果）

3.2.1 硬件成果：

硬件方面，成功集成 RK3588 主控、OV13855 摄像头和 HLK-LD2451 雷达，完成多接口接线与驱动适配，实现摄像头视频流与雷达数据的稳定采集。优化了分辨率、帧率和串口通信，保障多传感器同步，提升了系统的实时性和可靠性，为智能车载终端应用奠定坚实基础。硬件正反面实物图如 19、20 所示。



图 19 硬件反面图



图 20 硬件正面图

3.2.2 软件成果；

软件方面，完成了摄像头与雷达数据采集、YOLO 行人检测 AI 推理、3D 可视化渲染及全屏 UI 交互等核心模块开发。实现了多线程高效调度与异常容错，系统可独立打包部署，支持桌面快捷方式和一键安装。整体架构模块化，便于功能扩展和后续维护。软件 UI 界面如图 21、22、23 所示。



图 21 软件主菜单

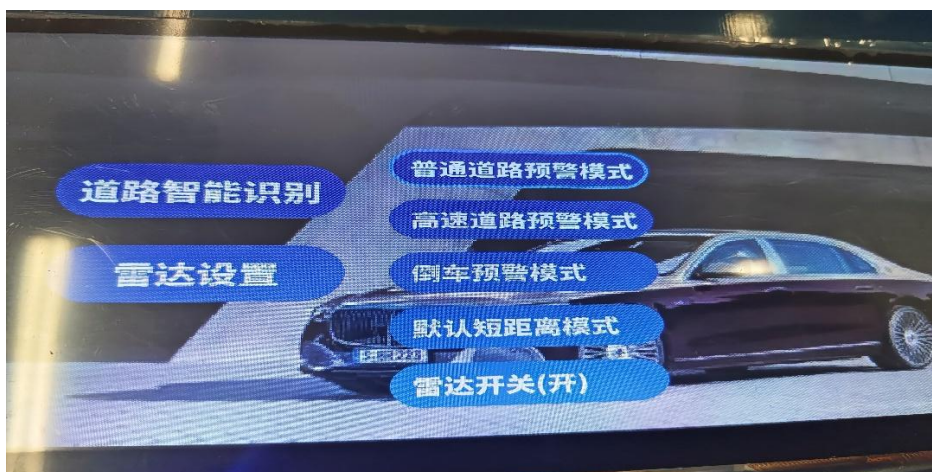


图 22 雷达模式选择界面



图 23 智能车载终端运行界面

3.3 特性成果

功能实地测试：

【纯视觉检测】如图 24、25、26、27、28



图 24 车辆识别远距离

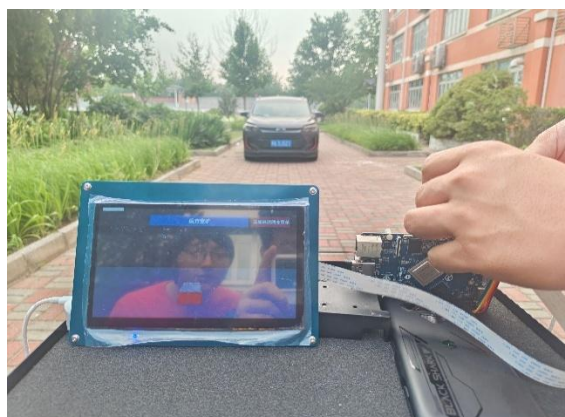


图 25 车辆识别中距离



图 26 车辆识别近距离



图 27 车辆及行人检测（右）



图 28 车辆及行人检测（左）

【视觉及雷达测试】如图 29、30



图 29 雷达测试以及远距离车辆、近距离行人



图 30 超远距离雷达测试及汽车、行人

以上部分为视觉及雷达检测车辆和行人的实地测试，确定能够准确检测出车辆和行人，并且正确标识在 UI 界面上，因为不可抗力导致存在 10%左右误差（拍摄时为大风天气摄像头以及雷达无法固定，使用手动固定，导致摄像头及雷达晃动速度存在 10%。）因为光线变化原因导致视觉测距存在 1%-5%左右误差

误差说明：如图中检测距离为 19m，实际测量距离约为 21.5m，图片仅展示部分数据，实际误差经多次测量计算得出。

第四部分 总结

4.1 可扩展之处

4.1.1 AI 模型与推理后端的灵活切换

现状：主程序通过 ultralytics.YOLO 加载 YOLO 模型，推理部分(detect_vehicles 等)与 3D 渲染、UI 解耦，模型路径和推理方式可配置。

可扩展性：可无缝切换或集成多种 AI 模型（如 YOLO、SSD、Transformer 等），支持不同推理后端（如 RKNN、ONNX、TensorRT、OpenVINO 等）。只需在模型加载和推理接口处增加适配层，便于后续升级算法、支持多模型协同检测，甚至实现模型热插拔和在线更新。

4.1.2 第三方外设与车载总线协议扩展

现状：当前系统主要与摄像头、雷达等基础外设交互，未涉及车载 CAN、LIN、以太网等总线协议。

可扩展性：可通过增加 CAN/LIN/以太网等总线适配模块，实现与整车 ECU、ADAS、仪表、T-BOX 等车载设备的数据互通。例如，读取车辆速度、档位、转向角等信息，或将检测结果反馈给仪表/中控，实现更深层次的车载系统集成。只需引入相应的总线库和协议解析模块，主程序即可通过统一接口管理多种车载外设。

4.2 心得体会

本次智能车载终端系统的开发，是一次软硬件深度融合的实践探索。项目以 RK3588（ELF2 开发板）为核心，集成了摄像头、雷达等多种传感器，结合 AI 视觉算法，实现了车辆与行人检测、雷达联动预警、3D 可视化等功能。

而整个过程中，硬件的选型、连接、调试和与软件的协同，更是项目成功的关键。

硬件选型与集成 首先，主控平台选择了瑞芯微 RK3588 ELF2 开发板。该板卡具备强大的 AI 算力、丰富的 I/O 接口和良好的 Linux 兼容性，非常适合车载场景下的多传感器接入和边缘 AI 推理。开发板自带 HDMI、USB、MIPI、UART 等接口，为后续扩展提供了极大便利。摄像头部分，初期采用本地视频文件进行算法开发和调试，便于复现和对比。后期正式集成了 OV13855 摄像头模块，通过 MIPI 转 USB 方案接入开发板。摄像头在系统中以/dev/video11 节点识别。实际调试过程中，遇到过摄像头无法识别、分辨率设置无效、画面花屏等问题。通过 v4l2-ctl 工具反复测试，最终将分辨率锁定为 640x480，保证了 YOLO 模型推理的实时性和准确性。雷达部分，选用了 HLK-LD2451 短距离雷达模块。该模块通过串口/dev/ttyS9 与开发板通信。实际接线时，遇到过串口线序接反、波特率设置不符、串口权限不足等问题。通过查阅资料，使用 dmesg、ls /dev/tty*等命令确认设备节点，并在/etc/udev/rules.d/中配置串口权限，确保雷达模块能够被正常访问。雷达的多种工作模式（普通道路、高速、倒车、短距离）通过串口指令配置，需严格按照协议发送命令，调试时用串口调试助手反复验证。

硬件与软件的协同调试 硬件调通后，软件部分的开发与调试也紧密围绕硬件展开。摄像头采集通过 OpenCV 的 `cv2.VideoCapture` 接口实现，需确保摄像头节点和分辨率参数正确，否则会导致采集失败或帧率异常。雷达模块则通过 Python 的 `serial` 库进行串口通信，主程序中独立封装了 `radar_module.py`，实现了雷达的连接、配置、数据接收和回调机制。为保证数据实时性，雷达数据接收采用多线程异步处理，避免阻塞主 UI 线程。在 3D 检测界面，摄像头视频流和雷达数据实现了同步融合。摄像头负责前方目标的检测与测距，雷达则补充了距离和角度信息。两者数据通过全局变量和回调函数实时更新，并在 OpenGL 渲染界面中以状态条、提示框等形式直观展示。硬件数据的稳定性直接影响到 UI 的流畅性和系统的可靠性。

遇到的问题与解决方案 整个开发过程中，硬件相关的问题尤为突出。首先是驱动和兼容性问题，部分摄像头和雷达模块在不同内核版本下表现不一，需要针对 RK3588 平台选择合适的驱动和固件。其次是电源和信号干扰，开发板和外设需独立供电，避免因电流不稳导致设备重启或数据丢失。再次是接口冲突，开发板的部分串口、USB 口与系统默认设备冲突，需在设备树或系统配置中手动调整。此外，硬件与软件的联调也暴露出一些边界问题。例如，摄像头采集帧率过低会导致 AI 推理卡顿，雷达数据丢包会影响预警准确性。为此，软件层面增加了多重容错和异常处理机制，确保即使部分硬件临时失效，系统也能平稳运行。

收获与展望 通过本次项目，深刻体会到硬件基础对于智能系统的重要性。只有硬件选型合理、连接可靠、调试充分，软件功能才能稳定发挥。未来可进一步扩展多传感器融合、车载总线对接、云端数据同步等功能，打造更智能、更可靠的车载终端系统。