

定位通信协同一体的智慧网联车系统

霸气的名字贰叁叁；董锦达；鞠昕妍；王卓扬；

指导老师：丁克蜜 尉进

摘要

随着制造业向智能化的发展，自动导引运输车（AGV）被广泛应用在了工厂和仓库的无人化操作下的物流运输和物品分拣等环节。目前，大多数 AGV 系统采用中心化的管理方式，例如通过连接局域网中心服务器等网络实现中心化控制和任务分配。然而，研究表明，这种中心化的协作模式存在较高的通信延迟和响应时间，特别是在高负载状态下，这种延迟会显著影响生产效率。[1][2]

局部去中心化的策略可以使 AGV 在保持全局集中式架构的基础上，通过车对车（V2V）通信直接从临近 AGV 获取信息，从而在局部问题解决上展现更优的性能。局部去中心化不仅提高了系统的灵活性，而且保留了全局优化的鲁棒性。[3]

本项目采用局部去中心化的协同通信策略实现邻近 AGV 的协作通信，通过使用超宽带（UWB）定位技术确定 AGV 的位置，实现邻近 AGV 之间的去中心化网络连接，在跨过中心服务器的情况下实现车辆间的直接通信，实现车-车协同场景，采用人工势场算法[4]对车辆进行起点到终点的布局，以提高 AGV 协同工作的响应速度。我们基于去中心化的网络通信模式，针对多辆 AGV 交汇时的避让问题进行研究，将处在交汇区域附近的车辆实时组成一个局部去中心化的网络，以寻求更高效的局部最优会车解，并检验去中心化车-车通信在协作避让中的效率。

第一部分 作品概述

1.1 功能与特性

通信：我们实现了去中心化的车与车分布式通信网络，车辆与车辆之间通过

这样的网络直接进行交流。

定位：采用 UWB 定位，我们实现了车辆相对于基站坐标系的笛卡尔坐标。

协同：通过对车辆设置任务（起点和终点），车辆能够对路径进行规划，并通过车与车之间的通信网络了解各个车辆的位置，完成车辆之间的躲避，成功到达终点。

1.2 应用领域

交叉口避让：在实际应用中，多辆 AGV 在交叉口交汇时，如何高效避让是关键问题。传统中心化管理因通信延迟，无法实时高效避让，导致交叉口拥堵甚至碰撞，影响生产效率。局部去中心化策略通过 V2V 通信实时共享信息，使 AGV 在交叉口时能直接协商，决定优先级并规划路径，减轻中心服务器负担，提高复杂环境的应对能力。

动态任务分配：在制造车间或仓库中，任务的动态分配也是 AGV 系统面临的重要挑战。传统中心化系统需要将所有任务请求发送到中央服务器进行处理和分配，在任务密集时容易造成服务器过载，延迟任务响应时间。局部去中心化策略允许 AGV 根据实时情况自主分配和调整任务。

故障处理：在 AGV 系统中，个别车辆出现故障是不可避免的。传统中心化管理方式需要将故障信息上传至中央服务器，再由服务器调度其他车辆进行故障处理和接替任务。这种方式在信息传递和响应时间上存在明显的滞后。局部去中心化策略下，邻近 AGV 可以直接感知到故障车辆的状态，并迅速采取行动。

1.3 主要技术特点

定位：我们采用超宽带（UWB）定位技术来确定 AGV 的位置。具体为，采用两个基站，根据两个基站的相对位置和两个基站到目标小车的距离计算出小车相对于基站坐标系的位置。

通信：我们借助 UWB 模块，通过自建数据帧，完成车车之间、车与基站之间的通信。附录-附 1 中提供了我们的数据帧格式。

避障算法：为了更好的适应嵌入式开发平台，我们采用了人工势场法——

种低复杂度的算法来实现车-车之间的躲避。不同于 A*、弗洛伊德等传统路径规划算法，该算法不需要对路径进行复杂的全局规划，极大降低了车辆路径规划的算力。附录-附 2 中提供了我们的避障算法。

1.4 主要性能指标

我们的实验场地是一块长度为 4.5m，宽为 3.0m 的水泥地面。

静态避障实验：我们进行了 20 次静态障碍物躲避实验，其中 19 次能成功躲避障碍物。

动态避障实验：我们进行了 20 次针对动态障碍物的躲避实验，其中 17 次能成功躲避障碍物。

我们设置的通信频率为 10Hz，考虑到延迟、超时、丢包等因素，我们测得我们的实际通信频率为 8.47Hz。

表 1：性能指标

指标	参考值
实际通信频率	8.47Hz
静态避障成功率	95%
动态避障成功率	85%
避障算法复杂度	O(N)

1.5 主要创新点

一方面，不同于其它采用中心服务器连接小车的通信方式，本项目参考**通信感知一体化**的顶层设计思想，不使用激光雷达、摄像头等成本较高的传感器，不依赖中心化的服务器，仅利用超宽带（UWB）通信模组，采用**去中心化**的思路进行多车定位与车间通信，以实现多车运动控制及车-车智能避让的目标。

另一方面，本项目采取了一种**不需要提前进行路径规划**的算法，极大降低了车辆实时进行动态规划的响应时间，降低对处理器的需求，提高了车辆的运输效率。

1.6 设计流程

首先，我们使用 Robot Master 全向麦轮运动小车底盘，通过 CAN 总线通讯向车底发送控制信息，同时车底将控制信息中的速度控制信息转换为四个车轮的运动信息。

随后，我们使用 UWB 模组进行定位和通信，通过研究和编写基站和标签算法，实现基站和标签之间、标签与标签之间的通信，并实现定位任务和信息传递任务。

完成定位和底盘控制之后，我们根据定位数据，设计中央控制器，包括其与 UWB 模块的通信和运动底盘的通信，在中央控制器中设计避障算法并对底盘进行速度控制，实现了车车之间的协同作业。

第二部分 系统组成及功能说明

2.1 整体介绍

系统整体结构分为两个子部分：车辆（Vehicle）和基站（Anchor）。

系统包含两个基站，每个基站均为一个 DWM1000 模块。主要负责接收来自车辆上端 DWM1000 模块广播的 UWB 信号，并返回相应用于计算距离的时间戳。

每个车辆由四个基本模块组成，自上而下分别是 UWB 模块、中央控制器模块、底盘控制器模块、电机。UWB 模块负责与基站和其他车辆之间的通信，按照一定频率广播自己位置信息的同时，接收来自基站的时间戳和其他车辆的位置信息，并通过串口 USART 与中央控制器模块通信。中央控制器模块（STM32F405）通过内置算法计算车辆的速度，通过 CAN 总线将车辆的速度信息（ v_x, v_y ）传递给底盘控制器。底盘控制器模块（STM32F427）接收中央控制器模块的车辆速度信息，计算得到各个车轮的速度（ v_1, v_2, v_3, v_4 ），发送给电机驱动控制电机，最终控制车辆运动。

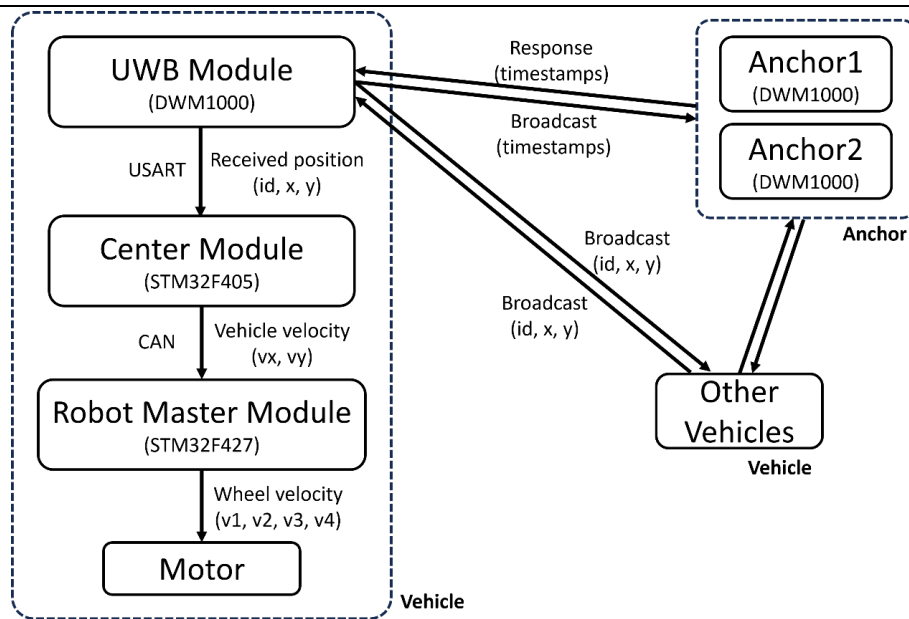


图 1：系统整体结构

2.2 硬件系统介绍

2.2.1 运动小车底盘系统

小车采用大疆 Robot Master 教学用车，主控采用 stm32f427 系列控制器，通过 CAN 总线与底盘车轮驱动器建立通信。小车为全向的麦轮小车。

2.2.2 定位器模块

本项目定位器采用超宽带(UWB)定位技术,使用超宽带收发模组-DWM1000 模块进行车辆定位。该模块为厂家封装后的超频宽带，我们使用该模块提供的 API 进行定位和消息收发操作。

2.2.3 中央控制器

我们的中央控制器采用 stm32f405RGT6 模块。中央控制器通过 CAN 总线与底盘控制器建立通信，通过 USART 串口与 DWM1000 通信模组建立通信。

此外，中央控制器通过蓝牙模块和 USART2 串口通信与实验者的个人计算机建立通信，便于调试输出。

2.3 软件系统介绍

2.3.1 软件整体介绍（含 PC 端或云端，结合关键图片）；

我们成功实现了车与车、车与基站之间的通信，并完成了小车的智能路径规划。此外，我们还编写了车辆位置可视化软件，实时可视化显示车辆的位置。

2.3.2 软件各模块介绍（根据总体框图，给出各模块的具体设计说明。从顶层到底层逐次给出各函数的流程图及其关键输入、输出变量）；

(1) UWB 模块（Anchor-UWB 和 Vehicle-UWB 模块的交互）：

UWB 模块被我们用作定位和通信的模块。

Vehicle-UWB 接受 Anchor-UWB 返回的时间戳，通过时间戳得到的 $t_1 - t_4$ 能够根据距离计算公式 $d = c_{light} \cdot (t_4 - t_1 - (t_3 - t_2))$ 计算得到 Vehicle 与两个 Anchor 的距离。随后通过余弦定理计算得到 Vehicle 在世界坐标系下的坐标 (x, y) 。Vehicle-UWB 模块每隔一段时间进行广播，以附录-附 1 中的数据帧格式广播自身 id、坐标、时间戳等信息。

Anchor-UWB 在接收到 Vehicle-UWB 广播的信号之后，将记录下 Vehicle-UWB 发送数时间戳、Anchor-UWB 接收到广播的时间戳、Anchor-UWB 发送回复消息的时间戳，并以数据帧的形式返回给 Vehicle-UWB。

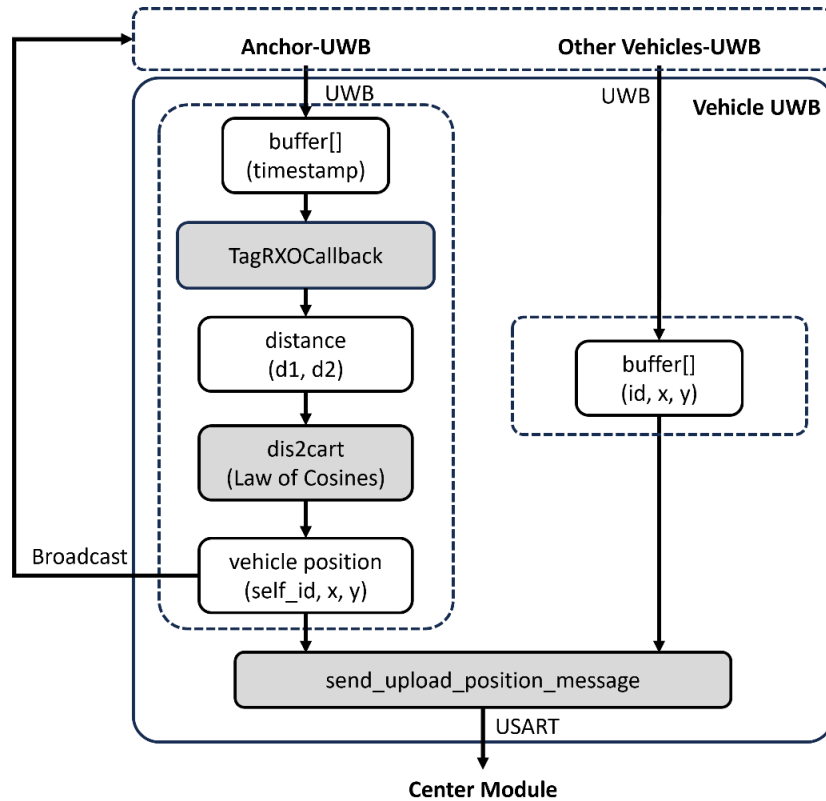


图 2：UWB 模块设计流程图

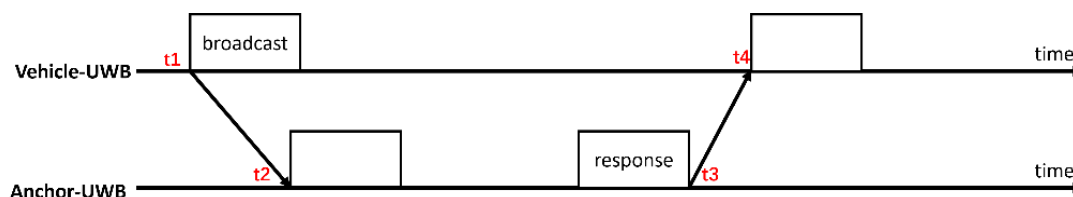


图 3: UWB 模块距离计算逻辑

(2) 中央控制器模块:

模块输入: 该模块通过 USART 串口中断函数, 接收来自 UWB 模块发送的数据帧。每个数据帧包含设备的 ID 和绝对坐标 (x, y) , 将接收的数据帧存入队列 S_Queue 中。

模块信息处理: 在同步进程 ostask_vehicle_controller 中, 从队列 S_Queue 读取数据帧 buffer[] 中的 id、x 和 y 信息, 并将其转化为浮点数格式, 存储在内存区域中。接着, vehicle_controller::tick() 函数会从内存区中获取相应的 ID 和位置坐标, 计算得到速度分量 vx 和 vy, 最终将结果存入 command 池中。

模块输出: 另一个同步进程 ostask_controller_module_port 通过读取 command 池中的 vx 和 vy 数据, 进行格式转换后, 将其通过 CAN 总线通讯发送给下位机 Robot Master Module。

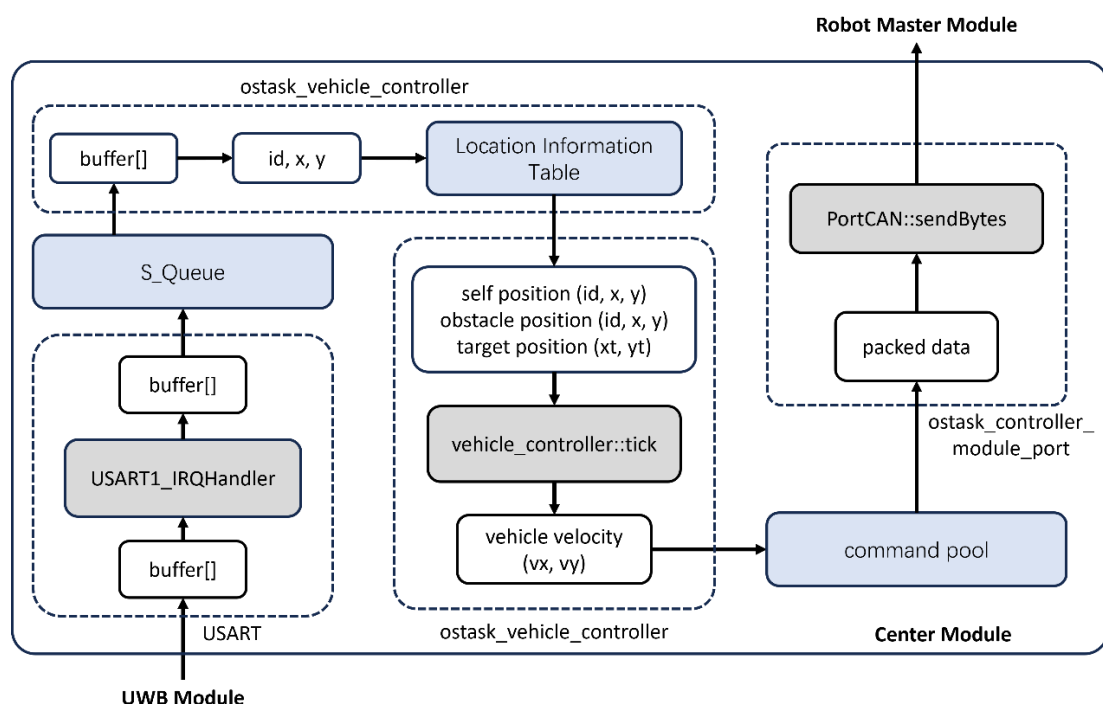


图 4：中央控制器模块设计流程图

(3) 底盘控制器模块

底盘控制模块我们使用了 Robot Master 大疆全向麦轮小车，通过 CAN 总线通讯向其发送经过中央控制器计算得到的 vx , vy , ω 控制信息，底盘将进行机器人逆运动学计算求解出四个麦轮的转速，从而完成小车运动状态的控制。

第三部分 完成情况及性能参数

3.1 整体介绍

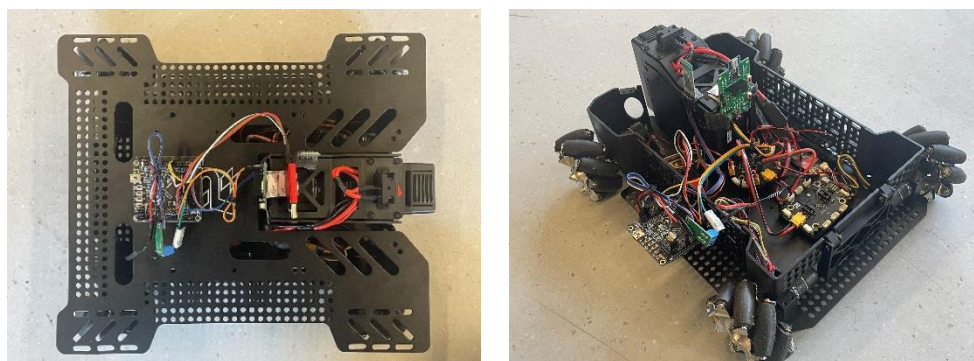


图 5：系统总体外观图。该系统为一个完整的包含了四个基本组成部分的小车系统子单

元，在多个相同的小车和两个基站结合的情况下能够组成车车交互系统。

3.2 工程成果

3.2.1 机械成果：

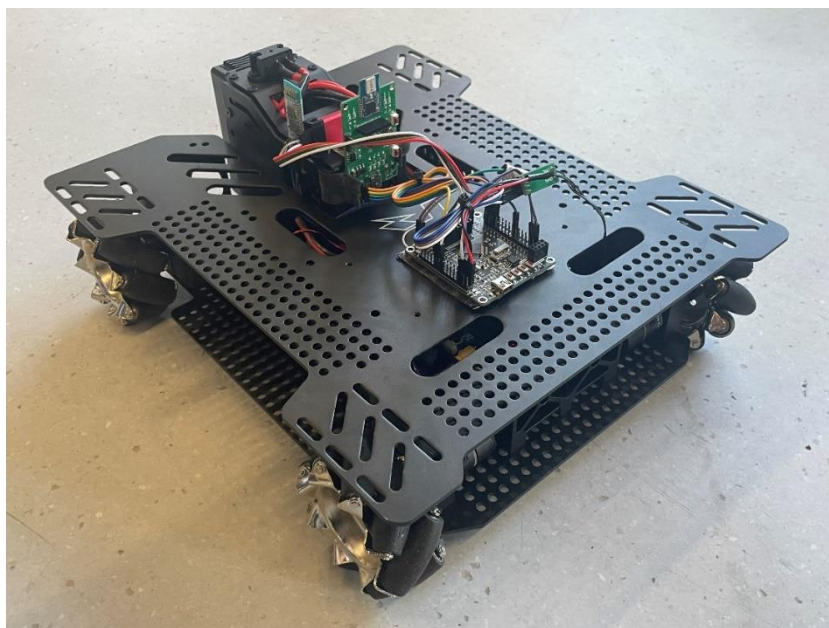


图 6：系统总体外观图。

3.2.2 电路成果：

系统内部接线图如下图所示：

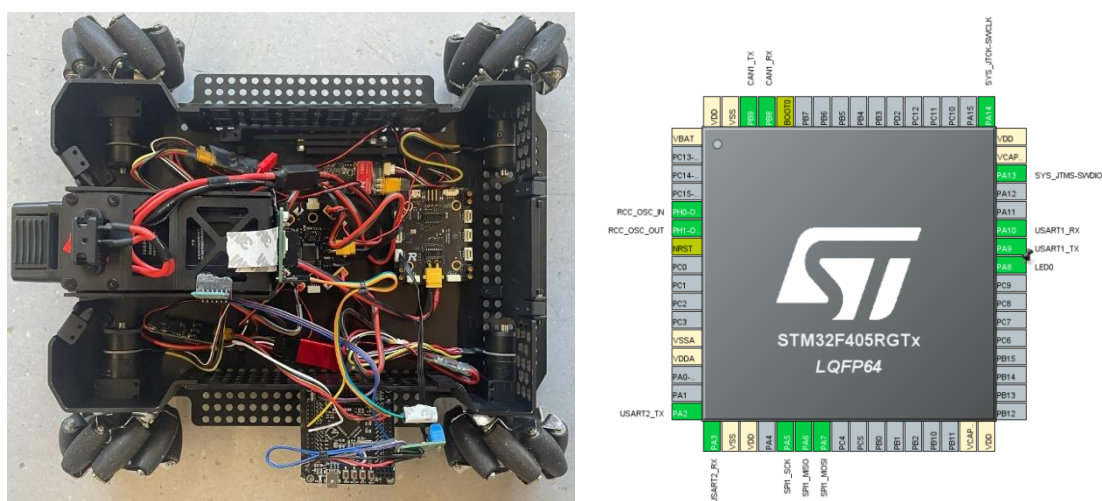


图 7：系统电路图和中央控制器开设的接口。

3.2.3 软件成果：

车辆运动全局可视化界面如下：

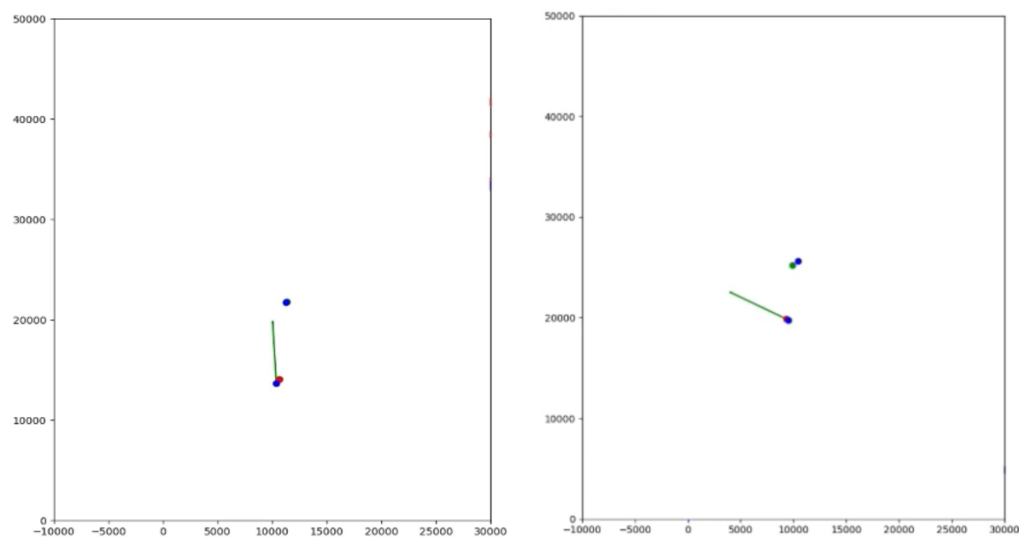


图 8：车辆运动全局可视化界面。

3.3 特性成果（逐个展示功能、性能参数等量化指标）

3.3.1 算法仿真：

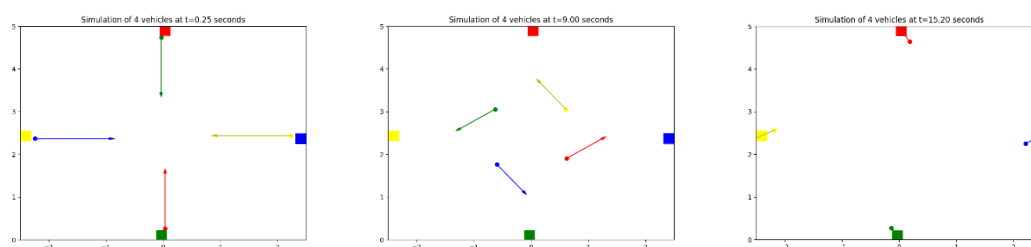


图 9：避障算法的仿真。圆点代表小车中心；对应颜色箭头方向代表小车速度，长度代表速度大小；小车终点位置为对应颜色的方块所在位置。从左往右第一张图为四个小车的出发状态，第二张图为运行过程中的避让状态，第三张图为即将顺利到达终点的状态。

第四部分 总结

4.1 可扩展之处

本项目成功实现了去中心化的车车通信，多个小车能够成功将位置信息发送给其它小车，在跨越中心服务器的情况下，通过分布式的网络通信实现群体智能规划。

本项目采用的分布式通信实现的群体智能规划，主要存在以下几个改进空间：一方面，我们的避障算法鲁棒性较低，将来的研究者可以采用更为复杂和有效的算法，提高小车避障效率，例如设置小车优先级等；另一方面，我们的通信模块较为简陋：我们采用的通信模块 DWM1000 是基于 stm32f103c6t6 的被高度封装的模块，拓展性较弱；由于实验条件有限，实验者只验证了两辆动态车辆之间的去中心化组网方式，对于多车之间的通信验证将在未来进行检验；此外，该模块所采用的开发板运算能力有限，在运算需求较大的场合下的表现还有待检验，未来的研究者可以更换运算能力更大的开发板。

4.2 心得体会

这是我们第一次在单片机的设计中进行合作。尽管是第一次合作，但我们分工明确，我们了解到，良好的封装和提前规划的接口设计将会使项目事半功倍，并且，项目组在完成项目的过程中，由于参加工业实习，有相当一段时间并不在同一座城市，调试难度相当大。因而我们通过提前设计模块间通信，提前拟定模块间的通信协议，提前大致确定模块的调用函数结构，让我们每一个测试成功的模块在最终组装的时候很快就派上了用场。

胜败乃兵家常事。项目进行中，我们也经历了诸多挑战和失败。调试过程中，项目组的遥控接收器因短路而发生爆炸，险象环生；因为缺乏有效的调试手段，项目组常常为简单的逻辑问题而调试很久；此外，项目组没有充足的经费报销渠道，项目组成员用爱发电，发挥人脉，完成了项目的搭建和测试……

先进工具通常是满足生产需求的重要途径。为了克服调试上的难题，我们引入了更先进的调试工具，尝试使用蓝牙模块和 USART2 串口通信与计算机建立通信，便于调试输出，并逐步完善了调试流程。

理论和仿真、仿真和实践之间存在着巨大的鸿沟，项目的进展过程中，我们深深地感受到了理论和实践的差距，实际的应用需要漫长的调试和更多情况的讨

论。例如，我们的调试输出和预想输出，在更换场地之后甚至也会出现不小的差距，我们将室外测试场景搬到室内之后，定位的准确度低了下来；我们在课桌上进行的定位有着极大的方差和不确定性，但是上机测试的时候却奇迹般地准确了下来。一方面，这某种程度意味着我们在嵌入式方面仍存在不足，促使着我们补足基础知识的同时加进理论和实践结合的进程；另一方面，这也告诉我们在设计系统的时候，应当进行更加周全的考虑。

总之，参加本次嵌入式大赛，我们收获满满。感谢全国大学生嵌入式芯片与系统设计竞赛组织委员会提供的良好机会，感谢指导老师对本项目的悉心指导。

第五部分 参考文献

- [1] Fanti M P, Mangini A M, Pedroncelli G, et al. A decentralized control strategy for the coordination of AGV systems[J]. Control Engineering Practice, 2018, 70: 86-97.
- [2] Li M, Guo B, Zhang J, et al. Decentralized multi-agv task allocation based on multi-agent reinforcement learning with information potential field rewards[C]. 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS). IEEE, 2021: 482-489.
- [3] Sun P Z H, You J, Qiu S, et al. AGV-based vehicle transportation in automated container terminals: A survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(1): 341-356.
- [4] RASEKHIPOUR Y, KHAJEPOUR A, CHEN S K, et al. A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles [J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(5): 1255-67.

附录

附 1：数据帧格式

以下为本次项目中采用的通信数据帧格式。

表 2：数据帧格式

位	0	1~2	3~4	5	6~7
说明	帧头	数据源 id	目标 id	CRC8	数据长度
位	8~11	12~15	16~23	24~25	26
说明	浮点数据	浮点数据	时间戳	CRC16	帧尾

其中，CRC8 为 uint8_t 类型的八位整型变量，数据源 id，目标 id，数据长度，CRC16 为 uint16_t 类型的 16 位整型变量，时间戳为 uint64_t 的 64 位整型变量。帧头为 0x5A，帧尾为 0x7F。

附 2：基于人工势场法的避障算法简述

首先，我们为小车设定终点坐标，我们让小车朝向目标方向直接运动，定义为基准速度。

通过通信模块，每辆小车可以获得其它小车的实时位置，因而我们在小车中，定义了一张车辆位置信息表以保存其它车辆的位置信息。当其它障碍车辆距离小车距离较小时，计算该障碍小车对自身的速度偏置。偏置的计算方法是：

$$\mathbf{v}_{bias} = \frac{v_k}{d} \mathbf{e}_d$$

其中， d 表示该小车和障碍物之间的欧几里得碰撞距离， \mathbf{e}_d 表示从障碍物指向该小车的单位方向向量， v_k 是一个速度标量常数。

最后，将所有的偏置速度施加到小车初始速度上，能获得小车的实时控制速度。

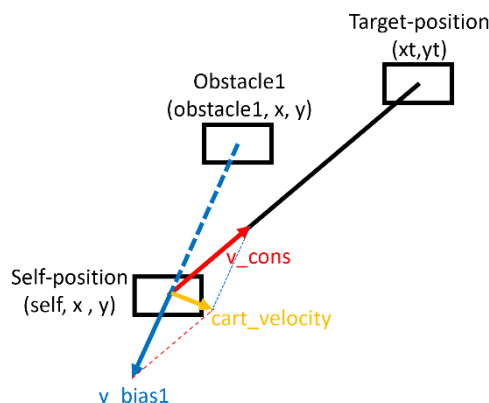


图 10：小车避障的控制速度计算过程。左下角的方块代表小车。红色箭头为根据小车当前位置和目标位置得到的基准速度；蓝色箭头表示障碍物给小车施加的速度偏置，从而让小车对该障碍物进行躲避；黄色箭头为最终该时刻小车的实际控制速度。

基于人工势场法的避障算法

输入：小车自身位置 p_c 、小车目标位置 p_t 、其它车辆位置信息表

输出：小车速度

- 1: 设定小车基准速度 $v = k(p_t - p_c)/\|p_t - p_c\|$ ， k 为给定速度追踪常数
- 2: **For** 获取其它车辆信息表中每个小车的位置信息：
- 3: **If** 障碍小车与当前小车距离较小：
- 4: 根据位置矢量，计算每辆小车对自身的速度矢量偏置 v_{bias}
- 5: $v \leftarrow v + v_{bias}$
- 6: **End If**
- 7: **End For**
- 8: **Return** 小车速度 v

算法 1：基于人工势场法的避障算法。