** **

**自主无人小车控制系统的实践方案设计**

**Report for Practice of Autonomous Car Control System**

|  |  |  |  |
| --- | --- | --- | --- |
| 学 院 | 电子与信息工程学院 | | |
| 专 业 | 自动化 | | |
| 组 号 | 第2组 | | |
| 学生姓名 | 贺兰山 | 学 号 | 1853760 |
| 学生姓名 | 蔡子坤 | 学 号 | 1854300 |
| 学生姓名 | 周熙哲 | 学 号 | 1850908 |
| 学生姓名 | 李达 | 学 号 | 1852916 |
| 指导教师 |  | 张 伟 |  |

2021 年 6 月 30 日

同济大学自动化专业综合设计与实践课程项目报告

Team Project Report

**<自主无人小车系统的实践方案设计>**

<2021.05.27>

目录

[摘要 3](#_Toc68814042)

[荣誉承诺 3](#_Toc68814043)

[0 名词术语 3](#_Toc68814044)

[1 课题意义 3](#_Toc68814045)

[2 目标和需求 4](#_Toc68814046)

[3 架构设计 5](#_Toc68814047)

[4 硬件设计 6](#_Toc68814048)

[5 软件概要设计 7](#_Toc68814049)

[6 软件详细设计 8](#_Toc68814050)

[7 演示、测试和效果评估 8](#_Toc68814051)

[8 工作任务分解、人员任务分派和进度与成本预估 8](#_Toc68814052)

[9 改进、增强与创新设计总结 9](#_Toc68814053)

[10请回答如下问题 9](#_Toc68814054)

[11 参考文献 9](#_Toc68814055)

[12 附件和提交物检查清单（Check List） 9](#_Toc68814056)

# 摘要

智能系统在悄然间已经融入了人们的生活，无论是在民用商品领域还是在工业生产领域，智能系统在越来越多的地方开始施展拳脚。自动驾驶技术是当今智能系统在民用领域的焦点之一，它的实现离不开软硬件的结合，离不开自动建图与导航（SLAM）算法的运用。无人车控制是入门控制最合适的训练项目之一，它包括了对系统硬件架构与软件架构的设计，包括了对无人小车运动最基础的控制，同时可进一步拓展学习SLAM算法，学习图像处理技术与计算机通讯技术，对于理解无人车系统的构成非常有帮助，同时又为将智能算法实践于无人车单元打下基础。

同时，网约车在当下已经十分流行，但是网约车受司机个人情况影响和地方位置影响较大，同时调度优化程度也不是很高，还有很大的优化空间。另外，独自乘坐网约车对乘客而言也有一定的风险，因此自主优化调度的无人网约车的研究具有很大的潜在价值。

本报告介绍了第2小组自主无人小车控制系统的设计方案，智能行驶主要开发的部分主要分为四个模块，即嵌入式部分通信模块，人机交互通信模块，地图管理模块以及图像处理模块，借用以上四个模块我们就可以实现一个自主优化调度运行的无人自主微缩滴滴系统，完成实现无人网约车的运行。硬件部分则采用Raspberry Pi 4 Model B为主控单元，编写并运行TinyROS操作系统，用STM32F103为电机控制单元，在客户端通过GUI界面操作，实现自主识别，建图，避障等功能，以网络通信的方式向小车发送控制命令以实现控制。此外还在gazebo中建立了仿真模型，实现了一个躲避几何形障碍物的简单避障功能，可在仿真模式下观察系统的运行情况并进行调试。测试结果表明，TinyROS顺利地完成了各个模块的调度；各个模块的功能顺利地发挥。

# 荣誉承诺

我承诺：尊重原始文章和代码作者的版权，所引用的网上的代码和文章等资源，均已在正文和参考文献一节中给出了引用。对第三方代码的使用，符合原始代码开源协议的要求。且本项目为我本人和小组成员完成，非本人和本小组成员的工作，均在正文中已经说明。

荣誉签名：\_\_\_贺兰山 蔡子坤 周熙哲 李达\_\_\_

日期：\_\_\_\_2021.05.27

# 0 名词术语

**TinyROS：**仿照ROS的“节点——话题”的结构编写微型机器人操作系统，实现任务调度、机器人控制；提供完备的命令行；具有动态性、易扩展性的特点。

**运动控制单元：**承担运动控制、底层传感器数据读取和各种执行机构控制。当承担安全相关

决策时，可独立决策。

**DMA**：Direct Memory Access，直接存储器访问，是所有现代电脑的重要特色，能够将数据从一个地址空间复制到另外一个地址空间。它允许不同速度的硬件装置来沟通，而不需要依赖于 CPU 的大量中断负载。  
**阻塞：**指调用调用某一个函数时，函数没有接收完数据或者没有得到结果之前，不会返回。  
**异步**：调用一个功能，不需要知道该功能结果，该功能稍后产生结果后会发出通知。  
**API：**Application Programming Interface，应用程序接口，是一些预先定义的接口（如函数、HTTP接口），或指软件系统不同组成部分衔接的约定，用来提供应用程序与开发人员基于某软件或硬件得以访问的一组例程，而又无需访问源码，或理解内部工作机制的细节。

**巡航：**指由多辆车组成的车链能够前后跟随完成规划路线的运动，并动态保持车间的距离。

**图像识别：** 通过摄像头判断图像中的二维码位置，并完成二维码的切割，扫描以及信息读取。

**小车定位：**通过图像识别二维码得到当前的位置，同时和UWP相结合得到小车的运动姿态。

# 1 课题意义

1.1 潜在市场和社会应用价值分析

随着计算机、智能化技术的开发越来越快，智能程度也越来越高，应用的范围也得到了极大的扩展。智能小车系统以迅猛发展的汽车电子技术为背景，涵盖了电子，计算机，机械，传感技术等多个学科。同时，当今机器人技术的发展日新月异，其应用于考古，探测，国防等众多领域。无人飞船，外星探测，智能化生产等等无不得益于机器人技术的发展。从某种意义上来说，机器人技术反映的是一个国家综合技术实力的高低，而智能小车是机器人的雏形，它的控制系统的研究与制作将有助于推动智能机器人控制系统的发展。

智能车又称为轮式移动机器人，是移动机器人的一种，是一个集环境感知、规划决策、自动驾驶等多种功能于一体的综合系统。智能机器人作为现代社会的新产物，是以后的发展方向，它可以按照预先设定的模块在一个特定的环境里自动的运行，可运用于科学勘探等，无需人为的管理，便可以完成预期所要达到的或更高的目标。智能机器人正在代替人们完成任务，凡不宜有人直接承担的任务，均可由智能机器人代替，可以适应不同环境，不受温度、湿度等条件的影响，完成危险地段，人类无法介入等特殊情况下的任务，例如中国最新发射的天问一号探测器的“祝融”号火星车，为人类对火星的探测做出了巨大的贡献。

同时，网约车在当下已经十分流行，但是网约车受司机个人情况影响和地方位置影响较大，同时调度优化程度也不是很高，还有很大的优化空间。另外，独自乘坐网约车对乘客而言也有一定的风险，因此，通过将智能小车和出租车相结合的自主优化调度的无人网约车的研究具有很大的潜在价值。此外，微缩滴滴系统可以全天候进行对用户的响应，切自助规划汽车调度和路线规划，一方面可以减少用户等车需要的时间，另一方面也可以在一定程度上减轻城市的交通压力，使得大家的出行更加的边界和快速，避免较大的交通拥堵，形成最大程度上的公共交通优化策略。

1.2 成本效益分析（或投入产出分析）

本课题需要7,300元，人工160小时，按目前开发人员的月薪20000元/月计算，总直接开发成本为14600元。推广与销售费用总计为20,000元，预期年销售量为3,000，单价为2,000元，年销售额为6,010,000元，相关专利收入为10,000元

列表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序 | 投入说明 | 预估  金额 | 产出说明 | 预估  金额 |
| 1 | 履带式坦克车 | 1000 |  |  |
| 2 | 树莓派4B开发套件 | 248 |  |  |
|  | STM32Nano开发板 | 128 |  |  |
|  | 陀螺仪JY61 | 38 |  |  |
|  | STM32微控制器程序开发，人工80小时，目前开发人员的月薪10000元/月计算 | 3000 |  |  |
|  | TinyROS程序开发，人工160小时，目前开发人员的月薪10000元/月计算 | 6000 |  |  |
| 4 | 推广费用 | 10,000 |  |  |
| 5 | 销售费用 | 10,000 |  |  |
| 总计 |  |  | 预计年销售数量3000，单价2,000 | 6,000,000 |
|  |  |  | 相关专利收入 | 10,000 |
|  |  | 30414 |  | 6,010,000 |

1.3 竞争对手及市场同类产品分析

|  |  |  |  |
| --- | --- | --- | --- |
| 产品型号 | 开发商 | 产品特点 | 对比 |
| 亚博 智能小车 |  | **产品特点：**  ⊙内置多种传感器，实时监测传感器状态，实时查看小车识别画面并控制  ⊙通过OPEN CV对图像扭曲矫正,提取ROI区域,通过模糊处理消除噪点,然后转化为二值化图像,通过对比图像中线的偏离值,使用PID算法对机器人移动方向进行校准,达到自动驾驶的效果。  **产品价格**：449元/个 | 该产品与本课题产品相比，视觉识别能力会更好，智能驾驶效果较好，但无法做到智能跟车，对于多人搭车情况处理不够好。 |
| 阿克曼 ROS小车 |  | **产品特点：**  ⊙语音模块，有声源定位、语音导航、语音召唤功能  ⊙激光雷达建图与导航  ⊙深度视觉跟随  **产品价格：**10000元左右 | 该产品功能较多，但价格太高，不适合广泛推广到网约车行业。 |

# 2 目标和需求

2.1 设计目标

模拟某城市基于出租车的交通系统，路人甲和一些朋友共十余人在路边叫车，要 赶往某体育馆看演出。它们通过 MiniDiDi 系统发出叫车请求，一共来了 2 辆车，然后 2 辆 车组成一个车链，向目的地进发。

2.2 需求

功能需求：

|  |  |  |
| --- | --- | --- |
| 编号 | 说明 | 类别 |
| A1 | **能够搭建仿真环境，在仿真中驱动小车并保留实验数据以便调试** | 必做 |
| A2 | **支持多车形成车链，首车具备高度自主的智能行为，其他车辆尾随。** | 必做 |
| A3.1 | 对单车智能进一步提升，增加基于图像的感知能力，完成环境感知，建图以及二维码识别定位。 | *选做* |
| A3.2 | 能够引入强化学习等算法 | *选做* |
| B1 | **调用PID算法实现无人车的速度控制** | 必做 |
| B2 | 融入多种传感器或改进算法实现对无人车的运动实现更精确的控制 | 选做 |
| C1 | **能够进行位置标注，帮助理解环境。** | 必做 |
| C2 | 能够进行地图管理优化，应对复杂情况。 | *选做* |
| D1.1 | 能够通过人机交互控制小车运动并调用算法 | 选做 |
| D1.2 | **增加服务中心，包含多车、多人和地图在内的交通系统进行动态优化调度（增强）** | 必做 |
| D2 | **实现小车与其他小车的联网交互以及远程调度** | 必做 |
| E1 | **典型功能：实现小车运动的自主避障** | 必做 |
| E2 | 其他典型功能（如语音控制小车等） | *选做* |

性能和可靠性需求（技术指标要求）：

|  |  |  |
| --- | --- | --- |
| 编号 | 说明 | 类别 |
| 1 | 小车的最快运动速度能达到0.5m/s | 必做 |
| 2 | 小车的最快运动速度能达到1m/s | *选做* |
| 3 | 每秒至少更新1次小车摄像头的视野情况 | 必做 |
| 4 | 小车能够基本准确地躲避简单几何体障碍物 | 必做 |
| 5 | 小车转向误差不超过2° | 必做 |
| 6 | 小车的转向误差不超过0.5° | *选做* |
| 7 | 小车的通讯速率不低于10KHz | *选做* |
| 9 | 前后车的距离始终保持在0.5~1m之间 | *选做* |

# 3 架构设计

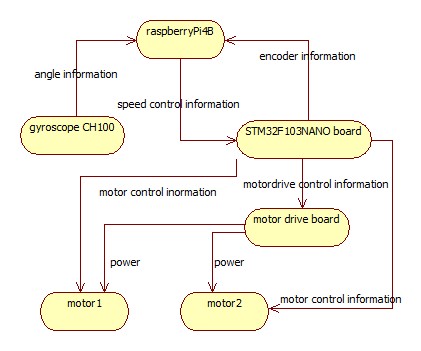


图3-1 硬件架构图

系统的硬件架构图如图所示。在小车上采用的主控制器为Raspberry 4B，下连STM32开发板和陀螺仪(JY61)。树莓派与STM32通过串口通讯控制驱动电机，与陀螺仪通过串口通讯获取角度数据。

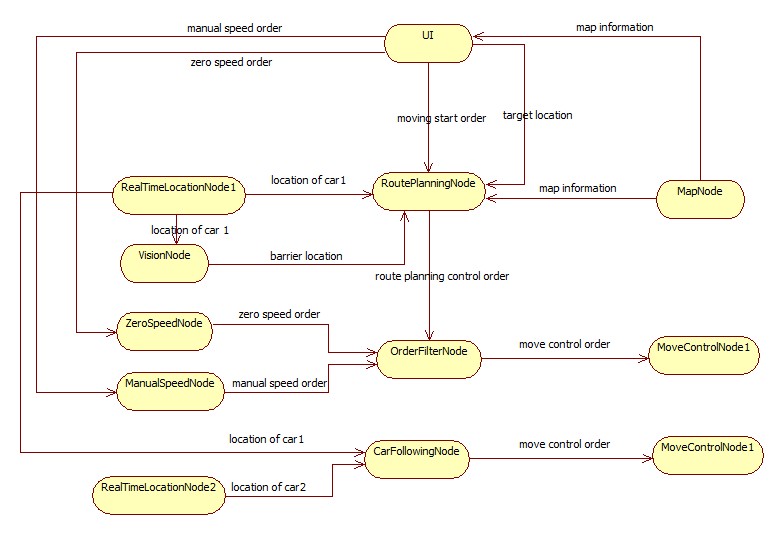


图3-2 软件架构图

系统的软件架构图如图所示。主要分为TinyROS操作系统、人机交互界面、STM32运动控制模块、地图服务器模块，路径规划模块和跟车运行模块。

在树莓派上的TinyROS操作系统负责联络组织各个模块，进行任务调度。TinyROS包含多个节点，其中比较重要的包括：MoveControlNode节点负责通过串口向STM32运动控制模块发送控制信号，控制小车运动，同时获取编码器数据；RoutePlanningNode节点负责根据地图、起点、终点等信息实现路径规划和运动规划；UI节点负责通过无线网络与电脑上的用户图形界面实现交互；MapNode节点负责生成地图并发送给路径规划节点。

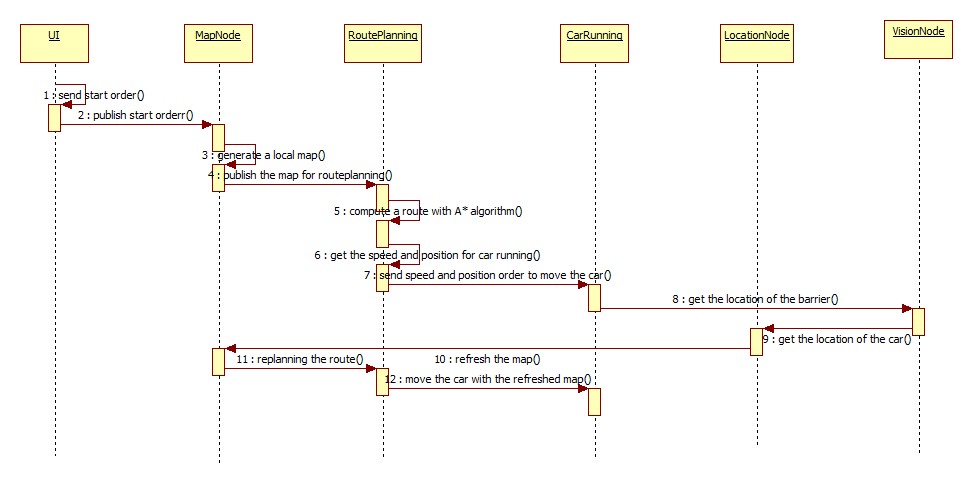
地图服务器运行在PC上，负责生成测试用的随机地图。

图形界面和路径规划节点运行在PC上，图形界面提供图形用户接口（GUI）。GUI程序为用户提供了小车操作界面和地图、路径观测窗口， 路径规划节点通过A\*算法进行路径规划，并转换成小车可执行的速度命令。

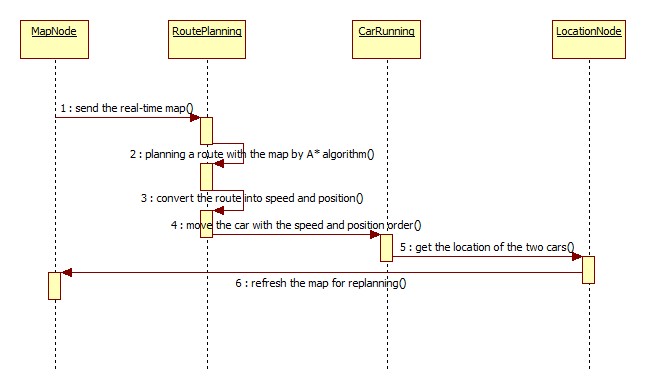
STM32运动控制模块是在STM32F103开发板上运行的程序，负责接收来自TinyROS的运动指令并控制小车运动，也负责读取编码器的数据并送回上层。

实时定位模块在树莓派上运行，通过三个uwb模块建立二维坐标，计算出当前两个小车的实时位置和小车间的距离，通过陀螺仪得出当前小车的朝向，最后将所有信息其发送给路径规划节点，帮助其实现跟车运行。

视觉感知模块也运行在树莓派上，通过调用两个摄像头实现深度测量，从而生成环境点云，构建出环境地图用于导航。



**自主运行小车UML时序图**

****

**跟车小车UML时序图**

**技术选型：**本项目用到了如下软件和工具：

|  |  |
| --- | --- |
| 软件工具名称和版本 | 用途 |
| STM32CubeMX、Keil5、STM32CubeIDE | 用于编写STM32上的C语言程序。 |
| 立创IDE | 用于绘制硬件连接图 |
| Visual Studio、VS Code | 用于编写TinyROS操作系统、图形界面等 |
| PyCharm社区版 | 用于调试和运行地图服务器 |
| Visio2019 | 用于绘制框图、UML图等图形 |
| Gazebo | 用于进行系统的仿真 |
| OpenCV | 用于进行图像的处理 |
| Qt | 用于实现用户操作界面程序 |

# 4 硬件设计

4.1核心器件选型

**4.1.1 STM32F103单片机最小系统**

STM32F103xx中等容量增强型系列产品供电电压为2.0V至3.6V，包含-40°C至+85°C温度范围和-40°C至+105°C的扩展温度范围。一系列的省电模式保证低功耗应用的要求。STM32F103RBT6系列拥有48个管脚，丰富的外设配置使之合于多种应用场合。

STM32F103x8和STM32F103xB增强型系列使用高性能的ARM® Cortex™-M3 32位的RISC内核，工作频率为72MHz，内置高速存储器(高达128K字节的闪存和20K字节的SRAM)，丰富的增强I/O端口和联接到两条APB总线的外设。所有型号的器件都包含2个12位的ADC、3个通用16位定时器和1个PWM定时器，还包含标准和先进的通信接口：多达2个I 2 C接口和SPI接口、3个USART接口、一个USB接口和一个CAN接口。

类似的STM32F103系列的MCU还有STM32F103ZET6，其拥有更大的片上存储空间和更丰富的外设，但在本项目中使用片上时钟产生PWM信号通过大功率直流电机驱动板进行控制，对于更丰富的外设并无要求，为节约成本，最大限度利用已有资源，最终选择使用板载STM32F103RBT6 的Nano开发板进行控制。

**4.1.2小车底盘：TS700坦克式履带小车**

小车车身采用碳钢全金属材料，耐磨耐用。其中坦克车车盘，驱动轮和承重轮均为全不锈钢材料，并在车盘涂黑色保护层。小车零件裸重为4.1kg，设计的最大载重可达10kg，能够通过Arduino驱动板或大功率直流电机驱动板实现控制。驱动电机采用大马力直流电机，并且可以根据需要的载重速度来选择电机转速，转速选择范围为5- 800转，供电电压为12V或24V。

**4.1.3电机：HM-37GB520**

HM-37GB520是微型直流减速电机，输入电压可为12V或24V，最大功率为10W，其转速比范围为30-400转。电机能够实现正反转，并且拥有霍尔编码器，通过上位机可以读取电机转速的数据。电机特性表如下所示。

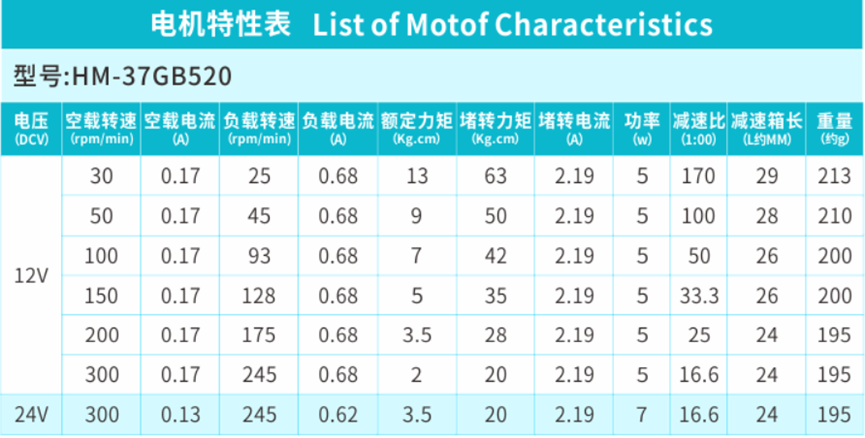


图4-1 电机特性表

**4.1.4大功率直流电机驱动板**

采用的电机驱动板为配套TS700坦克小车与HM-37GB520电机的驱动板，采用12V/24V直流电压供电，带有大散热片，该驱动器最大特点是易编程控制、正反转、强劲驱动与刹车功能、驱动有利、刹车迅速。驱动板采用完整的两片半桥驱动芯片＋极低内阻的N沟道MOSFET组成，使MOSFET的开关损耗降至最低，能够提高电源利用率。

**4.1.5 树莓派Raspberry Pi 4 Model B**

Raspberry Pi 4 Model B是流行的树莓派系列中最新产品电脑。它提供了突破性的处理器速度和多媒体增长。与上一代相比，CPU和GPU的运行速度都有所增长，内存扩大，且同时保留兼容性并降低了功耗。对于终端用户，Raspberry Pi 4 Model B也提供了可与入门x86级PC系统相媲美的桌面性能。

该产品的主要参数包括高性能64位四核处理器，拥有两个micro-HDMI端口，支持分辨率高达4K的双显示器，高达4K60帧的硬件视频解码，高达4GB的RAM，双频2.4/5.0GHz无线局域网，蓝牙5.0，千兆以太网，USB3.0和PoE功能。

**4.1.6 摄像头模块OV7725**

ATK-OV7725-V11是ALIENTEK推出的一款高性能30W像素高清摄像头模块。该模块采用OmniVision公司生产的一颗1/4英寸CMOS VGA (640\*480)图像传感器，OV7725。ATK-OV7725 模块采用该OV7725传感器作为核心部件，采用标准的SCCB接口，兼容IIC接口，支持VGA、QVGA，和从CIF到40\*30的各种尺寸输出，支持RawRGB、RGB(GBR4:2:2，RGB565/RGB555/RGB444)，YUV(4:2:2)和YCbCr(4:2:2)输出格式，集成有源晶振和和FIFO (AL422B)， 集成FIFO芯片(AL422B)，任意一款MCU都可控制该模块和读取图像，具有自动黑电平校准(ABLC)和自动带通滤波器(ABF)等图像处理功能，高灵敏度、控制所需电压低，适合嵌入式应用。

4.2电路图//硬件连接图

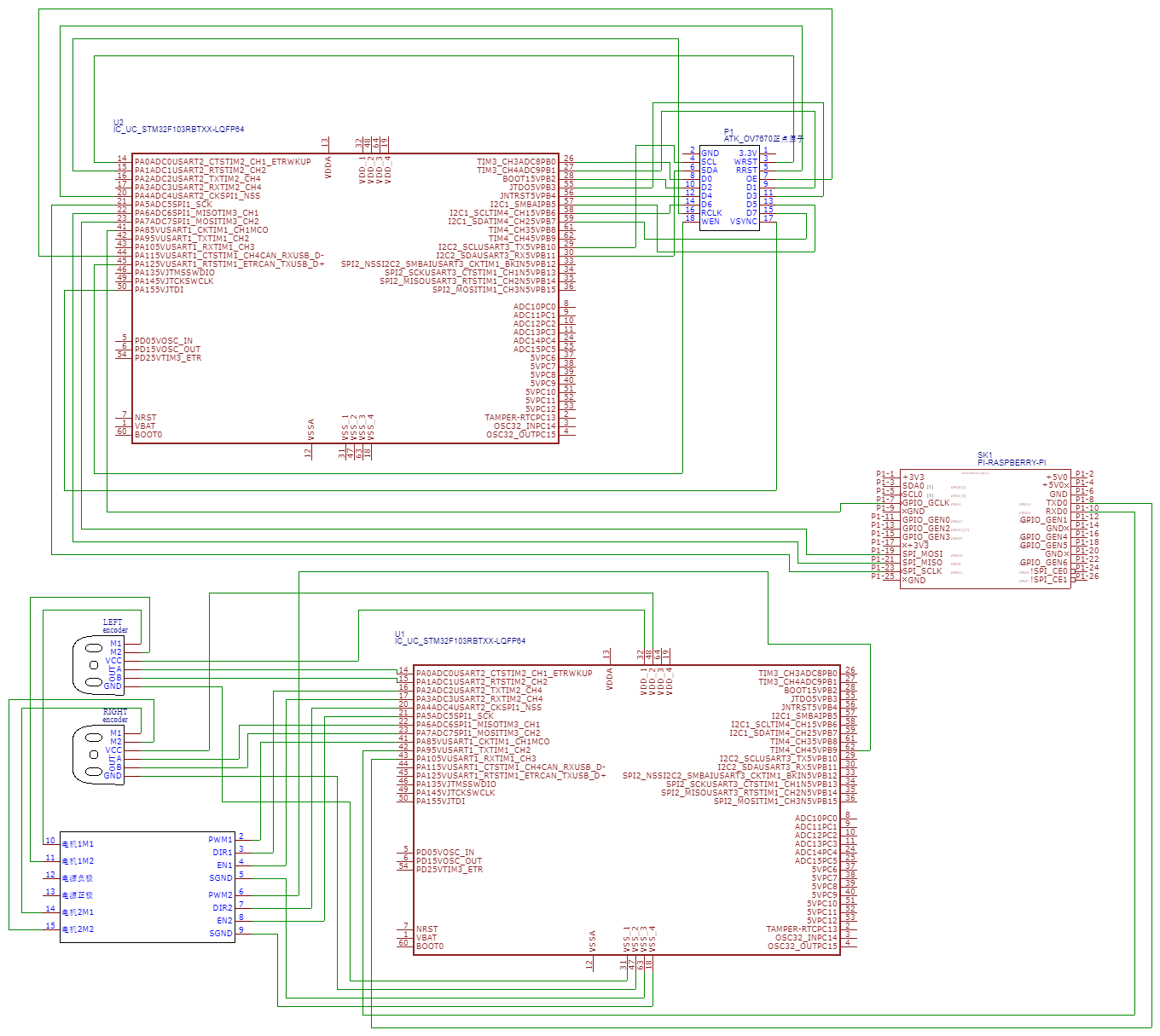


图4-2 硬件电路原理图

4.3 BOM清单（原材料清单）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 器件标号 | 厂家标号 | 主要技术指标 | 采购价 | 数量 | 总价 | 供货商  (URL) |
| U1 | STM32F103RBT6 | 128KB FLASH、20KB SRAM、3个通用定时器、1个高级定时器、1个 DMA 控制器（共 7个通道）3个串口、1个 RTC、2个看门狗、1个Systick定时器、1个 USB、51个通用 IO 口等。 | 178RMB | 1 | 178RMB | [正点原子](https://item.taobao.com/item.htm?id=575415395242&spm=2014.21600712.0.0)（淘宝） |
| 无 | HM37GB520 | 输入电压直流12V/24V，功率可达10W，转速比30-400转，可调速，可正反转 。 | 42RMB | 2 | 84RMB | 深圳市深联科技有限公司：[Hello Maker官方旗舰店](https://detail.tmall.com/item.htm?spm=a1z10.5-b-s.w4011-22721747414.84.98d05be9svINhP&id=619962005409&rn=ab33c0e43581a69ae0356016d977d197&abbucket=14) |
|  | 大功率直流电机驱动板 | 单路额定电流20A，工作电压8-30V，控制信号电压3.3-5V，每路控制信号电流mA，支持占空比范围0-99%，工作温度-10-80℃。 | 128RMB | 1 | 128RMB | 深圳市深联科技有限公司：[Hello Maker官方旗舰店](https://detail.tmall.com/item.htm?spm=a1z10.5-b-s.w4011-22721747414.69.747b5be9bJIn5w&id=619060201248&rn=db78e85c16a10714e2baa657c96b39f7&abbucket=14) |
| SK1 | Raspberry Pi 4 Model B | CPU：Broadcom BCM2711，1.5 GHz，64-bit，4核心，ARM Cortex-A72 架构，  无线网：b/g/n/ac 双频 2.4/5 GHz 蓝牙5.0  17个GPIO接口，4个UART接口，4个I2C接口，采用5 V USB-C 输入，或 GPIO 端口输入，待机 600mA（3W），满负荷 1.25A（6.25 W） | 230RMB | 1 | 230RMB | 开发套装：[耀麦旗舰店](https://detail.tmall.com/item.htm?spm=a230r.1.14.1.2533261e1h1GR5&id=601506908618&ns=1&abbucket=20) |
| 无 | OV7725 | 标准SCCB接口，兼容I2C接口，集成有源晶振与FIFO芯片，方便MCU控制，支持VGA等各种图像尺寸输出，支持RGB等多种图像格式输出，具有自动黑白校准等图像处理功能。 | 118RMB | 1 | 118RMB | [正点原子旗舰店](https://item.taobao.com/item.htm?id=561343242672&ali_refid=a3_430582_1006:1103191143:N:cBt%2F4rzNPwk3xMfoCkn0yA%3D%3D:47856b4d3424a013d22b949a463bf964&ali_trackid=1_47856b4d3424a013d22b949a463bf964&spm=a230r.1.14.11#detail) |
| 无 | TS700金属坦克小车 | 具有碳钢车架和不锈钢驱动轮，滚动承重轮，采用锌合金履带，裸重为4.1kg，设计载重10kg，尺寸为360\*240\*100mm | 1788RMB | 1 | 1788RMB | [TS700 全金属悬挂减震履带式坦克车底盘 钢结构/轴承运行 越野避障全地形运行机器人小车配置动力电 套件1底盘配12V37电机含码盘【图片 价格 品牌 报价】-京东 (jd.com)](https://item.jd.com/10029398192433.html) |
| 总计 |  |  |  |  | 2526RMB |  |

4.4 模块清单及功能说明

|  |  |  |
| --- | --- | --- |
| 模块代号 | 模块名称 | 功能说明 |
| Power | 电源模块 | 提供稳压电源，满足各模块供电需要 |
| UART | 串行接口 | 与上位机进行数据传输 |
| GPIO | 通用输入/输出口 | 将数据传入电机驱动板 |
| MCU | 单片机最小系统 | 控制所有模块的协调运行 |

# 5 软件概要设计

5.1 模块清单及模块功能说明

|  |  |  |
| --- | --- | --- |
| 编号 | 模块名称 | 模块功能 |
| 0 | TinyROS | 一个通用的、动态的、易扩展的微型机器人操作系统。 |
| 1 | Master  （隶属于TinyROS） | TinyROS系统的核心，负责节点的调度。提供了节点和话题添加、运行等方法。提供了动态加载程序集的方法。提供了完备的命令行系统。 |
| 2 | Message  （隶属于TinyROS） | 用于定义用户想要在节点间发布和订阅的消息类型，增加消息的选择灵活性和多样性。 |
| 3 | Node  （隶属于TinyROS） | 用来实现节点与节点之间的 通信，完成了节点的实现方法，可用节点进行发布话题和订阅话题，实现有节奏有规律的通信。 |
| 4 | Publisher  （隶属于TinyROS） | 用于节点发布话题，能够创建单个线程实现话题的发布。 |
| 5 | Subscriber  （隶属于TinyROS） | 用于节点对于话题的订阅，订阅相应话题后，每一次相应话题发布消息都能通过回调函数进行处理。 |
| 6 | RoutePlanningNode | Node的具体实现，实现路径规划功能。 |
| 7 | LocationNode | Node的具体实现，代表陀螺仪和uwb定位的节点，负责与物理层陀螺仪和uwb模块通信，获取角度和位置信息。 |
| 8 | OrderFilterNode | 筛选正确的运动控制命令，实现小车自主运动。 |
| 9 | ZeroSpeedNode | Node的具体实现，发布零速度，保证启动前小车不动。 |
| 10 | ManualSpeedNode | Node的具体实现，发布手动速度，用于小车调试。 |
| 11 | MapNode | Node的具体实现，用于生成地图和更新地图。 |
| 12 | CarFollowingNode | 用于跟车的运动控制。 |
| 13 | VisionNode | 通过摄像头进行环境建图感知。 |
| 14 | 用户图形界面 | 基于javascript的图形化的人机交互界面，作为客户端运行在PC端 |
| 15 | STM32运动控制模块 | 运行在STM32上的程序。 |
| 16 | UART  （隶属于STM32运动控制模块） | 通过串口与上位机通信。采用DMA的方式接收来自上位机的运动指令；将读到的编码器数据在中断中定期发送给上位机。 |
| 17 | 定时器  （隶属于STM32运动控制模块） | 触发定时中断实现调度；通过PWM生成功能控制电机速度；通过输入捕获功能对编码器进行解码，获取电机转速。 |
| 18 | GPIO  （隶属于STM32运动控制模块） | 输出电机的控制信号，包括驱动板的电源控制信号、电机的使能信号和方向信号。通过LED灯指示工作状态。 |

5.2 数据结构设计

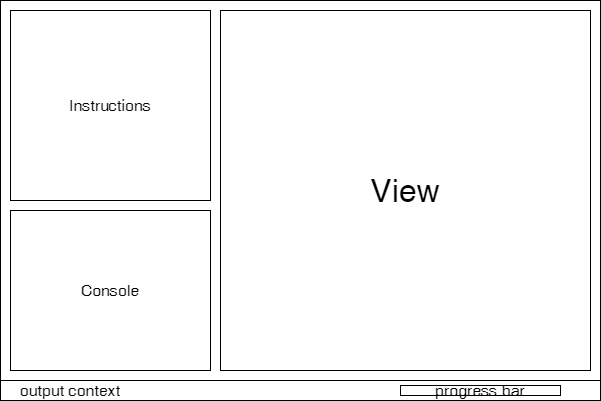
|  |  |  |
| --- | --- | --- |
| 编号 | 结构名称 | 结构功能 |
| 1 | Message | TinyROS内部消息的抽象基类 |
| 2 | SimpleObjectMessage | 简单对象消息，可用于传输简单对象。 |
| 3 | SimpleObjectArrayMessage | 可用于传输简单数组对象。 |
| 4 | CarStatusMessage | 包含小车位置、方向、速度、加速度状态的消息 |
| 5 | MapMessage | 包含地图信息的消息 |
| 6 | Vector3DMessage | 包含X、Y、Z三个float分量的消息。可以广泛用于各种实际情况，例如传递各种1至3维的坐标。 |
| 7 | StringMessage | 包含字符串内容的消息。可以用于各种命令的传递。 |
| 8 | DiffcarCourseMessage | 包含v，r两个float分量的消息，定义了小车的运动状态（线速度和转弯半径）。 |
| 9 | DiffcarVelocityMessage | 包含Left和Right两个float分量的消息，代表小车两个电机的转速。 |

5.3 人机交互设计（UI设计）

设计中的UI界面主要包括两个界面：命令行控制界面与按键控制界面。

**5.3.1 命令行控制界面**

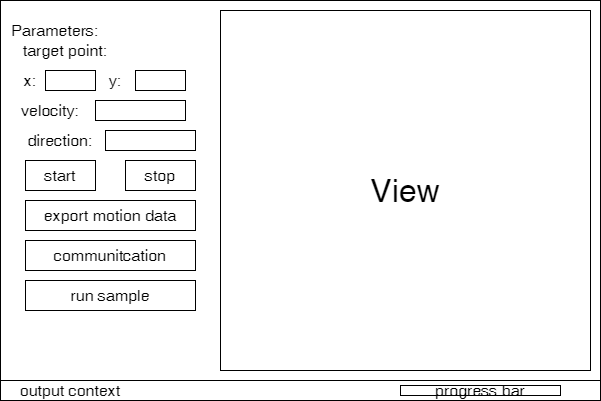
命令行控制界面设计如下图所示，其中Instructions面板用于显示可操控小车指令集的说明，Console是命令行交互界面，用于输入指令。Map是二维栅格地图显示界面，用于显示小车对环境的二维建图结果，可实时显示小车的位置。



**5.3.2 按键控制界面**

按键控制界面如下图所示。

对于小车运动的控制可通过target point实现。用户可在target point中填写小车移动目标点相对于当前的位置，期望移动速度以及方向角。按下start键，小车将根据输入的参数前往指定位置。按下stop键小车立刻停止运动。Export motion data按键可导出小车在运行过程中的位姿数据，用于详细观察小车的运动情况。Communication按键用于实现小车与其他模块的通信。按下按键后，小车将以一定频率向周围发送握手信号，一定时间内握手成功后会在output context中提示，否则将提示周围无人应答。Sample按键将展示程序控制小车的例程。



此外在起始面板中可以选择进入仿真或进入实物操控。进入仿真则加载Gazebo环境，进入实物操控则启动与小车运动控制模块的实时通信功能。

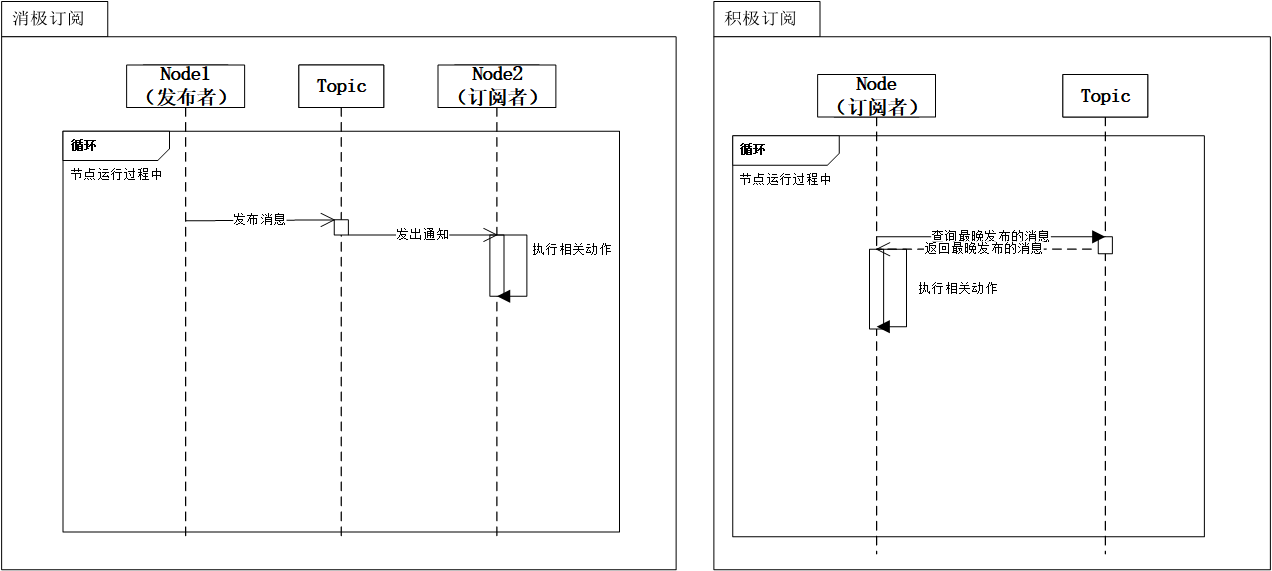
5.4 主要接口和通信协议设计

5.4.1 TinyROS内部模块接口与通信协议

接口：同一进程内部直接访问数据。

通信协议：采用话题——发布——订阅协议

通信的时序如下：



如图所示，节点之间通过话题传递消息，节点与节点之间是不耦合的。节点的订阅分为消极和积极两种方式，消极方式下，只有当其他节点向话题发布数据时，话题才会通知订阅它的节点，然后节点收到该消息执行后执行相关动作；积极方式下，无论其他节点何时发布话题，节点都会主动获取话题中的最新消息并且执行动作。

发布是异步的，发布者只是将消息发给话题，而不关心订阅者的接收状态。

5.4.2 Car模块与CarEntity模块接口与通信协议

接口：通用异步串行收发通信接口。

通信协议：8位字符数组char [8]



若方向为直行1，则忽略转弯半径（转弯半径=Inf）；

若转弯半径为0（原地转），则线速度是电机的线速度，否则线速度是两边电机的平均值。

5.4.3 ApplicationServer模块和ApplicationUI模块的接口与通信协议

接口：无线局域网

通信协议：Http协议，数据格式为json，具体设计如下表。



# 6 软件详细设计

6.1 TinyRos

6.1.1 概述

TinyROS是模仿ROS的节点-话题结构以及发布-订阅模型，编写成的通用的、动态的、易于扩展的微型机器人操作系统。TinyROS操作系统基于抽象类实现，实现了抽象类的类型都可以直接加入系统中使用。该系统具有动态加载程序集并且搜索其中类型的能力。该系统提供一个完备的命令行，可以发挥全部已经实现的功能。因此，该系统具有高度的动态性、灵活性、易扩展性，可以应用于多种场合。

6.1.2 类型设计

TinyROS面向对象开发，主要部分如下所示：

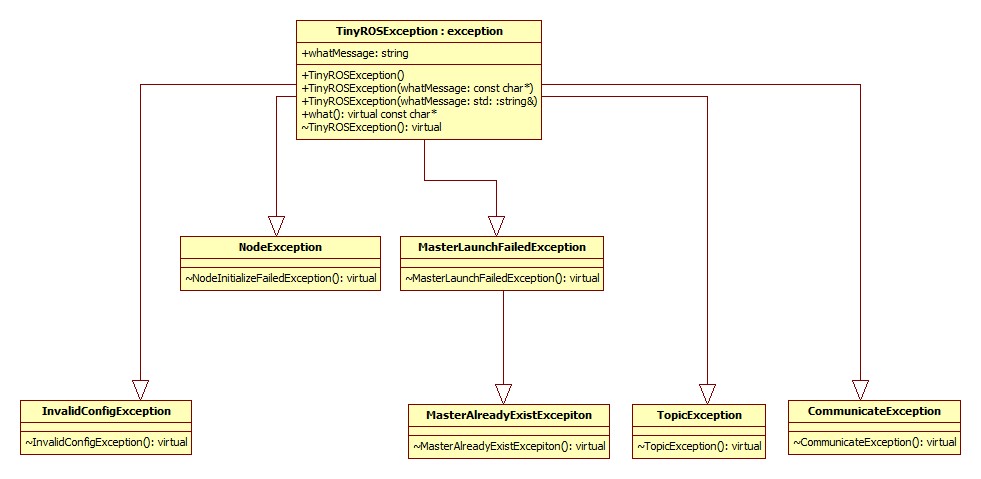


图3-3 Exception UML类图

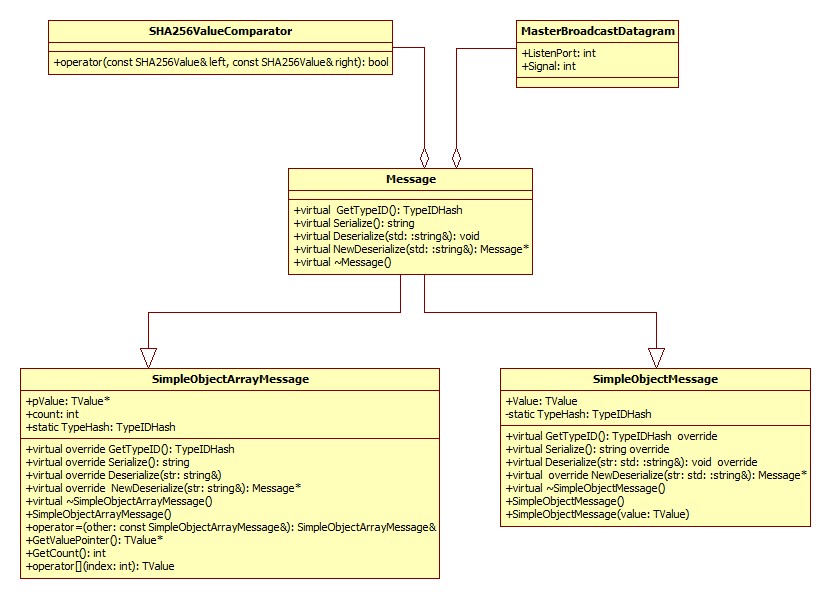
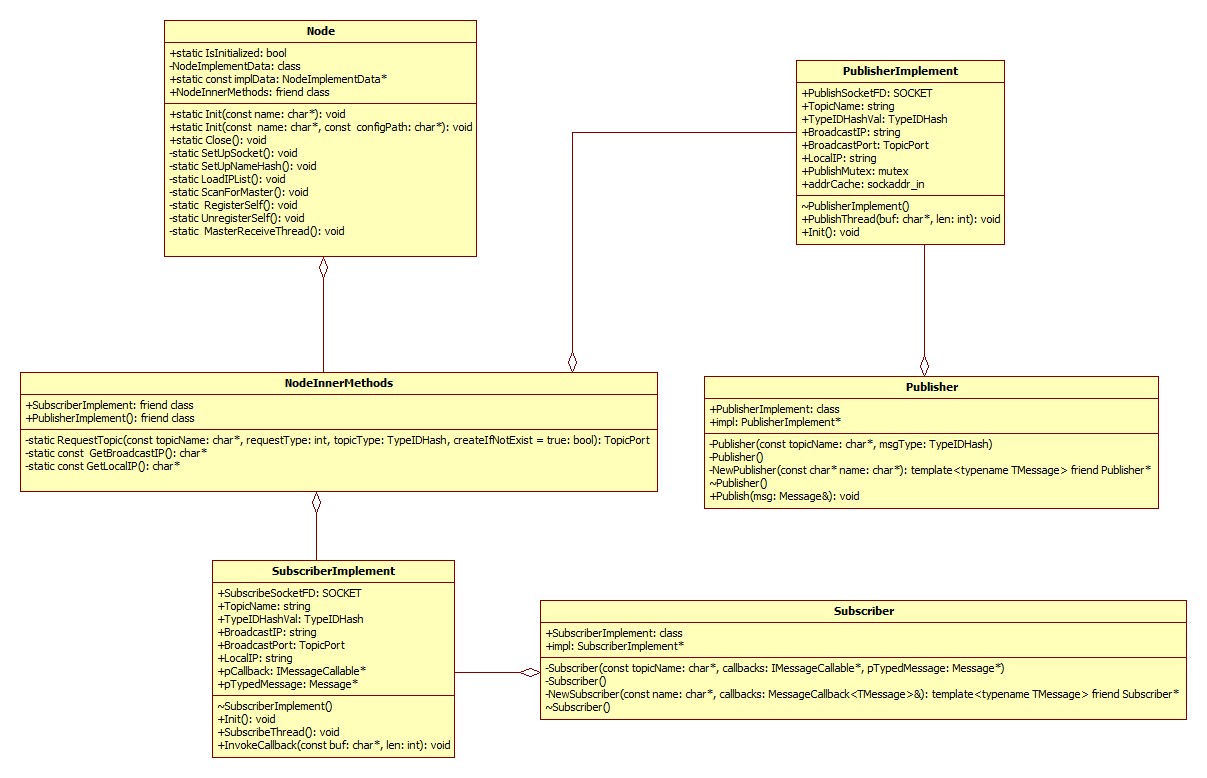


图3-4 Message UML类图

图3-5 Node UML类图

虽然看到图片中有很多的类型，但实际上，与系统核心运作直接相关的类型很少，如图所示：

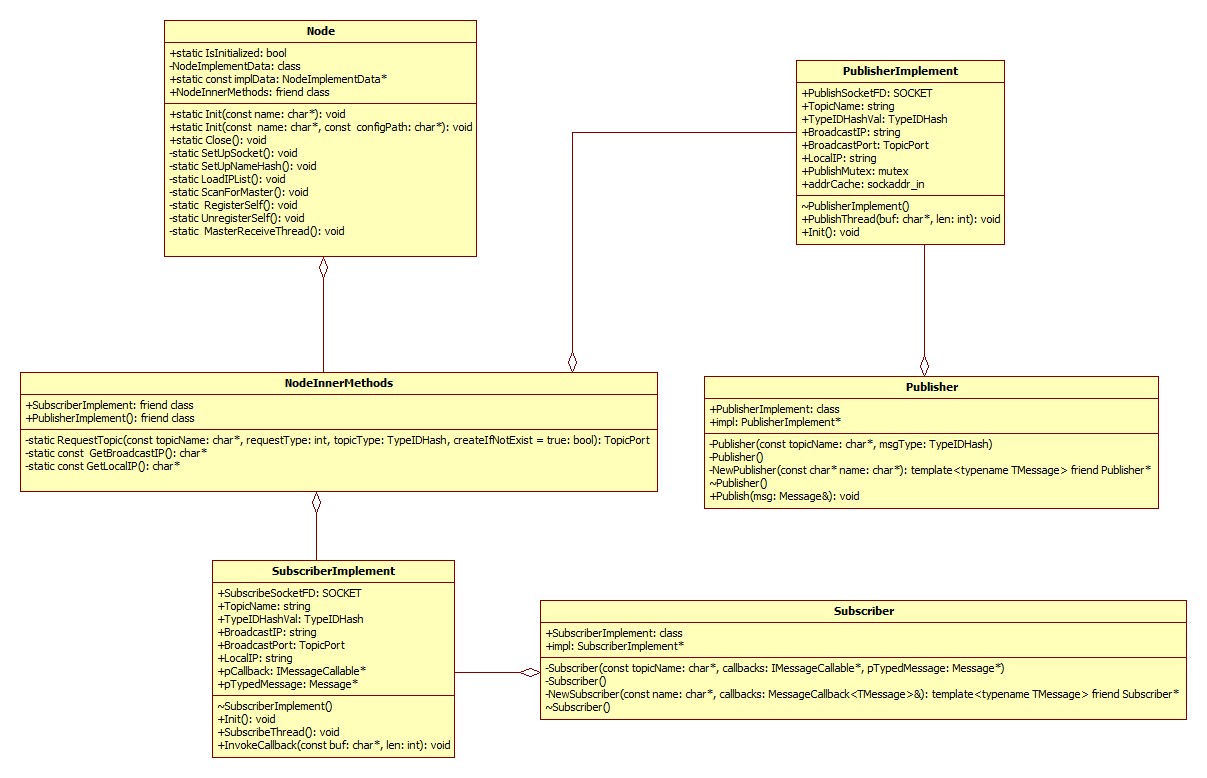


图6-2 TinyROS操作系统仅核心部分UML类图

可以看到，与系统运作直接相关的类型仅有6个，除了SystemCore外，都是抽象类或者泛型类。从各个类的各方法中也可以看出，操作都是基于抽象类的。这意味着，不论节点具体实现什么样的功能，在系统看来都具有统一的抽象行为，即初始化、运行和停止，使得系统具有高度的抽象性、一致性；同时也意味着，只要是继承自抽象类，实现了抽象方法的类型，就可以加入到系统中来，使得系统具有易扩展性。

本项目中实现小车的具体功能的各个节点或者话题，全部都是继承自上面的抽象类。

本系统中，还定义了丰富的异常类型，如图所示：

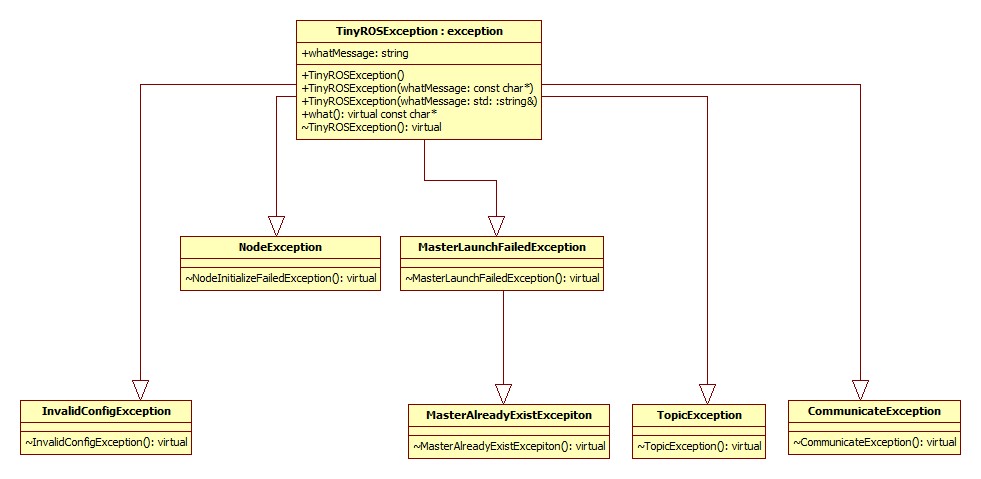


图6-3 TinyROS操作系统Exception部分UML类图

可以看到TinyROSException类继承自System.Exception类，随后，其余所有类都是从TinyROSException派生出去的。该图仅体现了类型的继承关系，没有标出类型的字段和方法。事实上，这些类型绝大多数也都没有定义新的字段或者方法，仅有几个类型增加了string类的字段。这些类型层次分明，适合进行异常处理，尤其是通过实现命令行功能的时候，需要处理各种各样的异常，并给用户相应的反馈。本TinyROS中，绝大多数其他异常都应被包装成这些自定义的异常。如果本系统后续应用于其他领域，进行扩展时，仍应遵守此原则。

**6.1.3 系统的运作**

节点和话题都由SystemCore来管理。用户（通过命令行）或者主程序创建节点和话题同时向SystemCore注册它们。之后，用户向SystemCore发出请求来运行或者停止节点，这一过程时序图如下图所示。

图示, 示意图

描述已自动生成

图6-4 TinyROS运行时序图

以上仍是基于抽象类的流程，因此不能体现出具体的功能。想要让系统具有具体的功能，首先需要实现节点类、话题类等，然后通过主程序或者在命令行中搭建系统。所谓搭建，就是将已经实现的节点、话题初始化，规定好节点和与话题的绑定关系，形象地说，就是连接起来，从而形成一个具有实际功能的整体。

本项目中，所搭建的系统结构如图所示：



图6-5 本项目搭建的系统结构

其中，椭圆形表示节点，圆角矩形表示话题；各个框内第一行是节点/话题的类型名称，第二行是节点/话题的名称。同一类型的节点/话题均可存在多个；节点/话题的名称可以任意取，但必须是唯一的。蓝色线表示订阅，橙色线表示发布。

各个节点的具体功能，已在前面的软件概要设计中阐述，这里不再赘述。

从中，可以看到TinyROS操作系统的一大重要特点就是低耦合性。节点和节点之间没有直接耦合，只能通过话题来传递消息。例如，路径规划节点并不能直接调用小车节点来控制小车，而只能发布速度话题，小车节点在接收到速度话题后再去控制小车。因此，节点是独立的，节点不会依赖于其他节点，只会依赖于相关的话题。节点从订阅的话题获取消息，但是并不关心话题的发布者；节点向话题发布消息的时候也不关心有哪些其他节点订阅了话题。

由于低耦合性，一个节点就是一个独立的模块，实现一种相对独立的功能。一个节点可能工作在不同的环境下，只要提供了相应的话题就能够运行。而将各种各样的节点、话题通过灵活的连接，就可以实现不同的系统功能。

以mux节点为例，它的类型是DiffcarCourseMuxByTimeNode，作用是订阅多个小车速度控制话题，这些话题具有不同的优先级，优先级高且消息有效的话题会被发布到最终的小车节点所订阅的速度控制话题里。这里的“有效”是时间在指消息生成后的3秒内（StandardMessage类包含时间信息，继承自它的类亦是如此）。之所以编写这样一个节点，是为了解决可能存在的冲突。在设计系统时，考虑到可能会有多个不同的来源想要控制小车，例如路径规划和远程客户端可能都要发布速度话题，但是小车节点并不关心发布者，因此可能存在冲突。如果改写路径规划的逻辑或者让小车判断发送者，或许可以解决问题，却破坏了模块的独立性。而现在，让mux节点订阅的3个话题，分别是appServer节点、planner节点和zeroSpeed节点在向话题发布（虽然实际的任务并不要求通过远程图形界面控制小车，appServer节点也没有真的发布消息，但是还是考虑这种假设情形），它们的优先级依次降低。这样，就实现了路径规划节点和远程图形界面都可以控制小车，并且远程图形界面的命令可以覆盖路径规划，而不会发生冲突，同时也没有破坏原有各个节点的独立性。而zeroSpeed节点的作用是定期发布0速度，保证了空闲时小车的速度为0。可见，通过灵活定义合适的节点以及连接关系，可以解决遇到的问题或者实现所需的功能。

**6.1.4 系统的异步调度**

TinyROS自然会面对多任务的处理。为了任务能够顺利并行执行，必须处理各种阻塞。任务中可能出现的阻塞，通常可以大致分为IO绑定和CPU绑定两类。IO绑定是指任务等待IO的完成才能继续进行，例如等待用户的输入或者等待请求的网络资源返回。CPU绑定是指某项任务消耗大量CPU资源使得其他任务无法执行。

为所有可能的阻塞分配线程固然可以解决，但并不是高效之选。从上述描述可以看出，IO绑定虽然造成阻塞，但是此时任务什么也不做，仅仅是等待，为它分配多线程，它也仅是在该线程上等待，是一种资源浪费。更为合理的方法是，将它和其他不会阻塞的任务一起放在同一个线程然后分配时间片轮询，也就是单线程异步。这样，让任务“并行”的同时节省了资源。

至于CPU绑定，该任务确实要花费大量时间来执行，从单个进程的角度来看并不能得到解决，但是从整个计算机的角度来看，多线程可以从全部进程中获取更多计算机的资源。因此处理CPU绑定用传统的多线程即可。

在TinyROS操作系统中，绝大多数的阻塞都是IO绑定，例如图形界面服务器节点和地图客户端节点的网络通信，以及各个积极订阅的Subscribers，都是IO绑定。仅有的可以称得上CPU绑定的地方是路径规划节点可能会产生大量计算（与地图情况有关）。因此，TinyROS使用异步编程，调用相关API，针对IO绑定，采用单线程异步的方式，节省资源，提高运行效率。

**6.1.5 TinyROS的扩展性**

由于SystemCore基于抽象类运行，因此具有很强的扩展性。任何开发者都可以基于现有的TinyROS进行开发，并在开发完成后加入现有系统使用，而无需改动现有内容。具体过程如下：①创建一个新的项目，将现有的TinyROS程序集（TinyROS.dll）设置为项目的引用；②继承自TinyROS中的Node、Topic、Message等抽象类来编写自己的程序，实现它们的抽象方法或者接口，同时实现自己所需的功能；③编译后即可使用。

这一过程中，无需改动现有的内容，也无需知道已有内容（如SystemCore）的内部实现。而随后想要调用新编写的内容，与调用原有的内容的方式完全相同。可见，TinyROS进行扩展十分方便。

同时，TinyROS具有动态加载程序集的能力，可以在运行时加载指定的程序集，查找其中的类型并且创建类的对象。这一过程相当于实现了字符串到类型的转换，提升了系统的动态扩展能力。这一功能需要通过TinyROS的命令行来实现。

**6.1.6 命令行与动态运行**

TinyROS提供一个完备的命令行工具。所谓完备，是指一切系统已经实现的功能都可以通过此命令行系统调用。此外，命令行还提供其他扩展功能，包括执行脚本。

TinyROS的命令系统如下：

|  |  |
| --- | --- |
| 命令格式 | 命令作用 |
| 节点类命令 | |
| node new <节点类型名> <节点名> [节点构造参数...] | 新建节点 |
| node run <节点名> | 启动节点 |
| node stop <节点名> | 停止节点 |
| node info <节点名> | 查看节点信息 |
| node list | 查看节点列表 |
| node execute <节点名> <节点命令...> | 由节点执行命令 |
| 话题类命令 | |
| topic new <话题类型名> <话题名> | 新建话题 |
| topic pub/publish <话题名> <消息构造参数> | 向话题发布消息 |
| topic delete <话题名> | 删除话题 |
| topic echo <话题名> | 侦听话题 |
| topic list | 查看话题列表 |
| 其他命令 | |
| load <程序集文件名> <命名空间名> | 加载指定程序集的指定命名空间 |
| execute <文件名> | 执行脚本 |
| config load <配置文件名> | 加载配置 |
| config save <路径> | 保存配置 |
| exit | 退出 |
| help | 显示帮助 |
| 备注：   1. 尖括号表示必须参数，方括号表示可选参数，带省略号表示由多个参数组成。 2. 各部分之间用空格分隔。 3. 红色高亮的命令表示关键命令，理论上仅通过这些命令就能够实现系统的功能。 | |

可以看到，这些命令已经涵盖了现有类型的公开方法，例如节点的新建、运行、删除、与节点命令的执行、话题的新建与发布等。因此，通过运行一系列的命令，就能够搭建出想要的系统结构。

更进一步，该命令行支持执行脚本。脚本是指由若干以上命令组成的文本文件，每行包括0到1个命令。使用execute命令可以执行脚本。执行脚本，相当于依次执行脚本中的每条命令。

由于命令行具有完备性，实现一个具体的系统结构，既可以通过源代码（编译执行），也可以通过命令行（脚本执行）。所谓编译执行，是指直接在源代码中创建想要的类，并且初始化、运行它们，编译后直接运行。而脚本执行，则是在源代码中仅仅启动命令行，其他什么也不做，运行程序之后，通过命令行的方式创建需要的节点并运行。这两种方式各有优缺点：编译执行的方式受到编译器的类型检查，更加安全，并且编译运行的效率更高，但是编译执行只执行某个确定的功能，每次修改都需要重新编译，同时需要使用者编写源代码；脚本执行可以自由决定系统组成，动态性更高，灵活性更高，修改脚本非常方便，而且使用者无需具有源代码的编写能力，仅需要使用上述命令即可。

二者也可以结合起来运行，先将一些固定的流程例如必要节点的初始化写在源代码里，并在源代码最后启动命令行，以便后续操作。

本项目中，也通过两种不同的方式实现了所需要的系统结构，它们是等效的。编译执行方式的主程序源代码如下：



图6-6 编译执行的版本的主程序源代码

可见，直接在源代码中创建了相关的类，然后初始化和运行它们。（最后也启动了命令行，虽然用不到。）直接运行该程序，就可以实现所需功能。该系统中有ApplicationNode节点（appServer），通过远程的图形界面进行操作，包括查看地图、设置起点和终点等。

而脚本执行的主程序源代码如下：

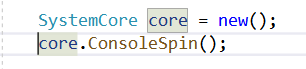


图6-7 脚本执行的版本的主程序源代码

可见，主程序仅仅是启动了命令行。接下来，需要运行程序，输入相应的命令。具体过程如下：首先，使用execute Demo.txt命令执行脚本Demo.txt，脚本内容如下所示：

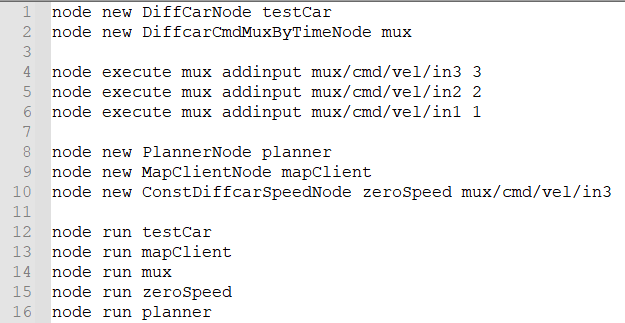


图6-8 Demo.txt脚本内容

可见，该脚本由一系列命令组成，基本流程与之前源代码中相似，都是新建并运行节点。不同的是，这些类型都是动态查找并建立的。此外，该脚本没有建立ApplicationNode节点，因此本系统不通过图形界面操作，而是直接通过命令行操作。

随后，手动依次运行下图中的命令：

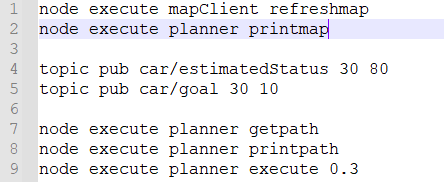


图6-9 xu.txt内容

之所以手动运行，是因为这些命令取决于之前命令的运行结果，需要使用者与之交互或者指定参数。对于命令的详细说明如下：

|  |  |
| --- | --- |
| node execute mapClient refreshmap | 该命令让mapClient节点向服务器请求生成一个新的地图。请求成功后会自动发布至话题，这样planner节点就会收到地图。 |
| node execute planner printmap | 该命令让planner节点在控制台上打印地图，从而使用者可以看到地图并且规定起点和终点。 |
| topic pub car/estimatedStatus 30 80 | car/estimatedStatus是表示起点的话题，后面的两个参数分别是地图的x、y坐标，范围0到99（地图大小是100\*100）。通过该命令即可设置起点，坐标需要在看过地图后根据情况给出。（如果坐标给到障碍里，后续命令就会提示找不到路径） |
| topic pub car/goal 30 10 | 与上一条类似，该命令设置终点。 |
| node execute planner getpath | 该命令让planner节点根据现有的起点、终点和地图规划处一条路径。该命令可能会运行较长时间。如果之前坐标给得不对，该命令会提示找不到路径。这时，可以重新运行前面两条命令，发布新的起点和终点再试一次。 |
| node execute planner printpath | 上一条命令找到路径之后，可以运行这一条命令将带有路径的地图打印在控制台上。也可以不执行。 |
| node execute planner execute 0.3 | 确认路径无误后，使用该命令即可让planner节点执行这条路径。后面的参数是参考速度，单位为m/s。该命令是异步的，会在后台运行，同时控制台可以继续交互。如要停止，可以使用命令node execute planner stop |

除了具有完备的功能外，命令行的交互也十分友好，有着丰富的提示，也有不同的颜色高亮。以下是一些示例：

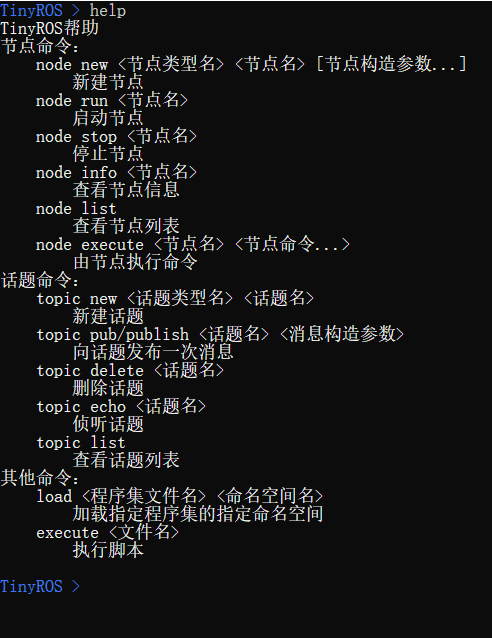


图6-10 help命令示例



图6-11 各种命令示例

可以看到命令行具有蓝色的“TinyROS > ”作为提示符，对于各种不同的内容都有相应颜色的高亮，对于不同的错误类型也有不同的错误提示。使用者可以清晰的了解到运行结果或者问题所在。

**6.2 电机驱动部分**

电机的驱动方法主要是通过调控PWM波占空比调控电机转速，并用STM32高级定时器实现霍尔编码器计数以计算电机速率和电机转向。所测算的电机速率采用平均移动平均法进行滤波，并根据误差采用PID控制器与前馈控制器结合进行调节。可以用如下的框图表示控制电机的系统。



图6-12 电机控制系统框图

小车转弯时的简化运动模型如图所示。记转弯半径为R，角速度为ω，履带小车两履带间几何中心距离为B，可以得到小车速度的计算式：

给定运动速度V与角速度ω或运动速度V与转弯半径R，在已知B的情况下可以确定两个电机所需设置的速度。在transform中根据该模型将设定速度转换为电机的设定速度。

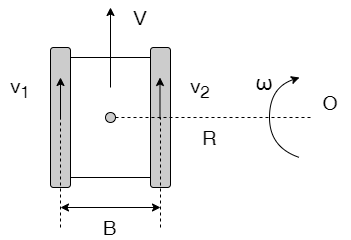


图6-13 坦克小车运动模型

STM32与树莓派通过串口通信时，发送的数据格式共7位，前3位代表速度，第4位表示转弯状态，后3位表示转弯半径。

**6.3 仿真部分**

首先需要创建世界环境模型，模型包括一个房间，房间内摆放着球形，圆锥形与长方形的障碍物。接下来创建小车模型，小车需要完成根据指定速度与方向角到达指定位置的功能。可通过自主编写gazebo插件控制小车。插件具有设置PID参数，设置速度以及拍照功能，并附有其他Gazebo规定的插件属性。

仿真部分的运行流程节点如下图所示。

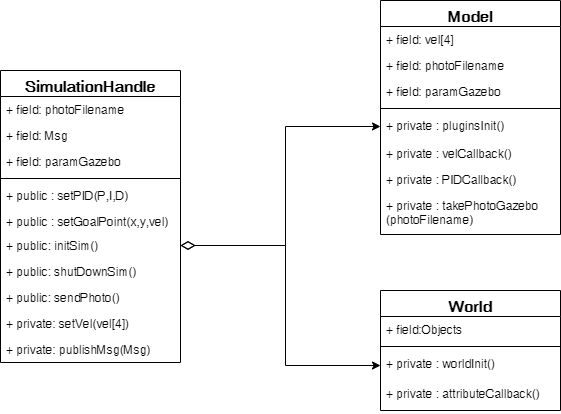


图6-14 仿真部分节点UML类图

实现小车的运动控制的过程为：调用setGoalPoint函数为小车设定目标点，可以附加运动速度和运动方向。该函数调用特定函数setVel，setVel主要实现的功能是：将期望的履带轮速度转换为期望的四轮车速度，调用publisher将目标控制速度以消息的形式发送至gazebo Model插件，最后观察运动结果。设置PID参数，控制小车拍照等功能均最终通过publishMsg对模型插件进行控制。该通信方法是基于Gazebo API的消息传送机制。流程图如下所示。

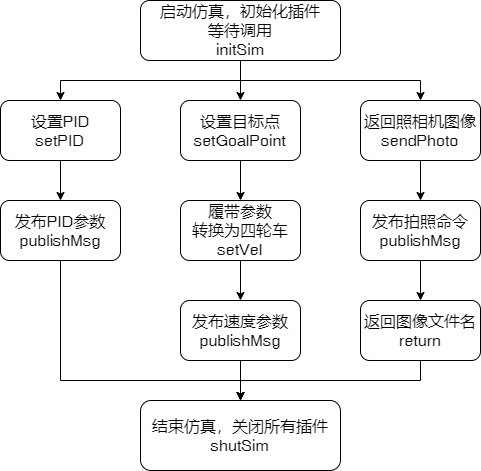


图6-15 仿真部分函数功能流程图

**6.4 GUI部分**

GUI部分主要实现电脑端与小车端通信，通过wifi与小车连接，将控制信息发送至小车。在GUI中主要模块与实现功能如下图所示。

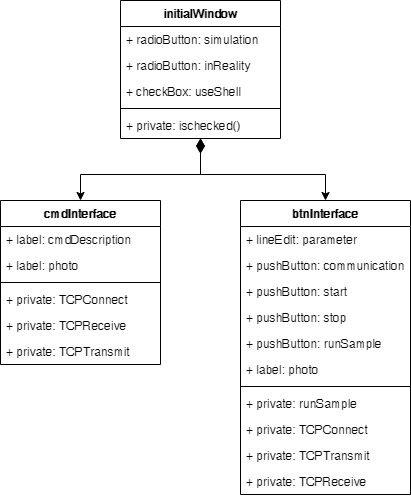


图6-16 GUI部分UML类图

进入主界面后，选择命令行或按键控制，选择是否运行仿真，之后进入控制界面。控制界面的操作在GUI设计中已进行阐述。对控制界面与小车的交互，主要通过WiFi收发信息完成。小车上的树莓派根据收到的信息指令控制底层。

**6.5 路径规划算法**

路径规划算法采用的是改进的A\*算法。A\*算法的表达式为



其中是总的价值函数，是当前位置到下一位置的距离，是下一位置到目标位置的预估价值函数。

对于网格化的地图，传统的A\*算法在做局部路径规划设定下一目标时，通常只关注邻域的8个位置。在本项目中，对A\*算法进行了改进，将设定下一目标的搜索范围拓展至半径为r的方格区域，如果这个方格区域没有障碍，则直接考虑区域的最外围。当半径r=1时即为传统A\*算法。设定下一目标的范围取周围没有障碍物的最大邻域。示意图如下。

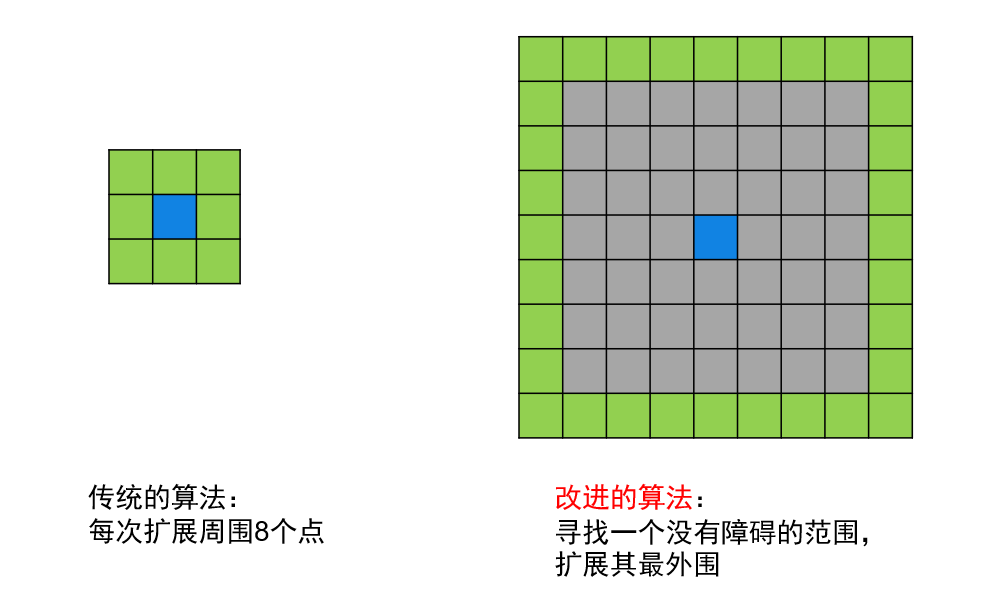


图6-17 改进的A\*算法示意

改进后的A\*算法能够行走的路线形式大大增加，从只能向45°方向（网格化地图8个相邻格的方向）变为可以走任意对角线的方向，使得行进距离缩短。同时，省略掉了中间不必要的点，使得规划出来的路径更加简洁，以便于后续处理。

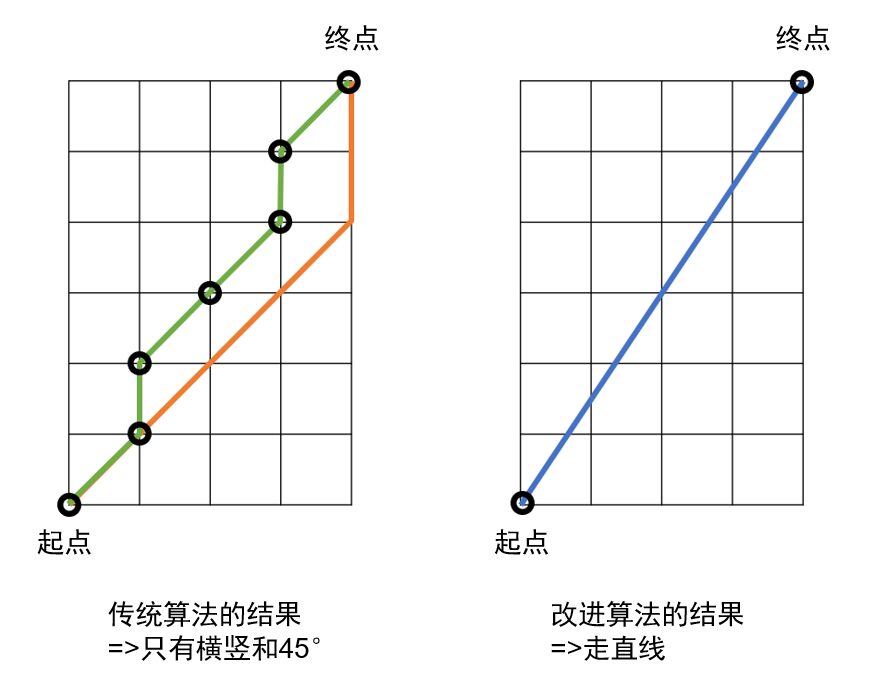


图6-18 改进的A\*算法的结果比较

# 7 演示、测试和效果评估

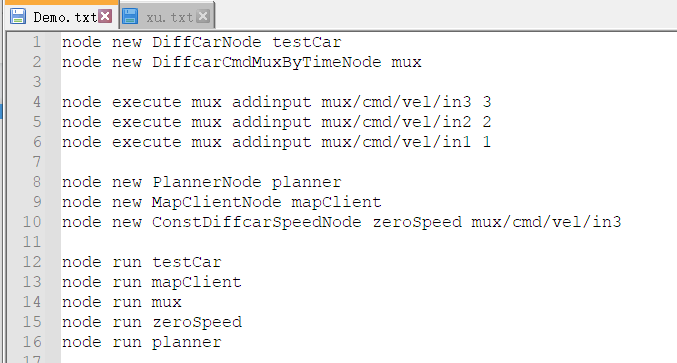
7.1演示/测试环境的搭建

（1）预装环境：在小车树莓派上安装.net5.0 runtime(使用微软官方install.sh安装)，在使用的电脑客户端上安装Qt框架。

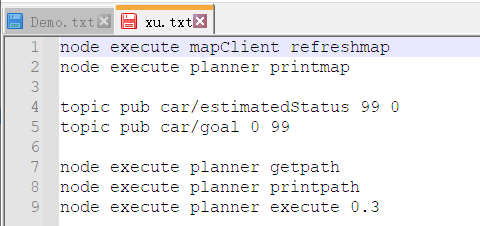
（2）配置小车树莓派的ssh与电脑客户端连接，即可通过命令行远程控制。

7.2操作步骤 / 实验过程说明 / 操作手册

通过ssh连接客户端连接小车后，首先进入可执行文件所在目录，通过Run.sh启动TinyROS程序，随后通过命令行输入以下命令：



完成对小车整个系统的搭建与启动。再依次输入以下命令（依次复制即可，每次只复制一条）



命令设置了小车运动的起点和终点，并指示小车规划运动路径，按照0.3m/s的速度运动。

完成对小车整个系统的启动后，也可通过启动客户端，在客户端中控制小车的运动。

7.3 测试结果及评价

演示过程与效果见附件视频。

测试结果表明，本项目编写的各个模块程序均能正常运行，并通过TinyROS系统完成了组织。底层控制使小车能够完成360度全方向的运动，且直线运动时的误差较小，控制效果良好。TinyROS保证了系统各模块完全分隔不互相干扰，且保证了系统的可拓展性，为系统将来引入新功能新组件提供了便利。系统能够通过命令行与客户端两种方式控制，既方便毫无基础的操作者操控系统，又为开发者提供了更接近底层的选项。

# 8 工作任务分解、人员任务分派和进度与成本预估

**绘制WBS图和甘特图**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序 | 任务说明 | 负责人 | 工作量（人天） | 成本预估 |
| 1 | 用STM32开发板连接电机驱动板，实现小车运动控制；  引入PID算法、卡尔曼滤波和陀螺仪，加强对小车的控制能力；  编写STM32与树莓派用串口通信的程序 | 蔡子坤 | 35 | 35\*100=2100 |
| 2 | 编写用户终端的GUI界面程序并实现与树莓派的网络通信 | 周熙哲、贺兰山 | 28 | 28\*100=1400 |
| 3 | 编写TinyROS操作系统，及一些具体功能的实现（树莓派上的串口通信、改进的路径规划、运动规划、地图客户端、GUI服务器），实现各个模块的整合。 | 贺兰山 | 35 | 35\*100=2100 |
| 4 | 引入路径规划算法 | 贺兰山、李达 | 14 | 14\*100=1400 |

# 9 改进、增强与创新设计总结

本项目的小系统参照了ROS进行设计，系统具有低耦合，可动态拓展的优点，为更有序管理系统各部分组件，以及后续拓展系统的功能提供了极大的便利。

本项目对传统的A\*算法进行了改进，使得搜索范围扩大，规划出来的路径更短、更直、更简洁。

# 10请回答如下问题

1. 本项目预期花费时间多长？实际花费时间多长？为什么出现差异？如果差异较大，今后你打算如何在项目管理中减小这种差异？

本项目预期花费两个月，实际花费三个月。出现差异的主要原因在于：

（1）前期设计不完备，对各功能的实现方法没有把握，查找资料不充分；在开发过程中时常发现更好的设计，使得模块之间要重新设计配合。

（2）开发进度掌握不当，一些功能未能按照预期时间点实现，开发管理能力有待加强。

吸取以上教训，今后在项目管理中应当注意：

（1）针对项目的各个功能模块，既要充分查找资料确定设计方案，又要保证各模块之间一定的可拓展性和低耦合性。既要追求设计一次到位，又要追求在设计不能一次到位的情况下仍能顺畅完成项目。

（2）密切关注开发进度，项目开始前就确定标志性时间节点。设计时间节点时要有一定预留量，以防出现差错难以调整。

1. 请对本设计方案做环境影响评价，包括但不限于生产制造环节的环境友好性、设备装置到期后回收的潜在污染等。

本设计方案硬件上采用通用性较强的嵌入式开发板，可以融入多种不同的系统中进行使用。项目组所使用的小车也可作为其他课程设计的材料，较好地实现了资源节约的目标。

1. 如果本项目实际上线运行或者销售，它对于社会、健康、安全、法律、文化、可持续发展的影响如何？（选择自认为最相关的一方面简要回答即可）

本设计方案对于自动化专业的学生，或任何对嵌入式系统有兴趣的学生均有所助益。通过让学生自行进行项目管理和团队合作，自行查找资料完成功能实现，对学生的能力锻炼有较大的作用。本项目中运用的专业知识也基本属于自动化专业或嵌入式相关的基础技术，通过完成该项目有助于开发者熟练掌握相关技能，为今后进入实际项目开发打下基础。

# 11 参考文献

无

# 12 附件和提交物检查清单（Check List）

|  |  |  |
| --- | --- | --- |
| 序号 | 说明 | 是否已提交 |
| 1 | 课程设计报告，其中应包含本小组全体同学学号和姓名（即本文档） | √ |
| 2 | 用于演示的PowerPoint幻灯片文件，其中应包含本小组全体成员姓名和学号 | √ |
| 3 | 报告和演示PPT中用到的视频Video文件 | √ |
| 4 | 硬件设计：电路图设计源文件，或提供本项目所用全部硬件的连接说明 |  |
| 5 | 硬件设计：要求用开发工具软件可直接打开，以及导出的pdf、jpg/png等格式的导出文件。 | √ |
| 6 | 软件设计：软件源代码工程及文件，要求用开发工具软件可直接打开 | √ |
| 7 | 最后生成的供烧写用的各种二进制文件 | √ |
| 8 | 其它希望提交争取加分的文件 | √ |