

毕 业 设 计（论文）

题目 基于ROS的移动机器人的设计与实现

学 院 计算机科学与工程

专 业 计算机科学与技术

班 级 118030702班

学生姓名 霍昊昌 学号 11823020232

指导教师 黄贤英 职称 教授

时 间 2022年6月

目 录

[摘 要 I](#_Toc103555357)

[Abstract II](#_Toc103555358)

[1 绪论 1](#_Toc103555359)

[1.1 选题背景及意义 1](#_Toc103555360)

[1.2 国内外研究现状 2](#_Toc103555361)

[2系统分析 3](#_Toc103555362)

[2.1 应用场景分析 3](#_Toc103555363)

[2.2 功能需求分析 3](#_Toc103555364)

[2.2.1 需求概述 3](#_Toc103555365)

[2.2.2 全向移动 4](#_Toc103555366)

[2.2.3 定位与建图 4](#_Toc103555367)

[2.2.4 路径规划与导航 4](#_Toc103555368)

[2.2.5 用户交互界面 5](#_Toc103555369)

[2.2 可行性分析 5](#_Toc103555370)

[2.2.1 技术可行性 5](#_Toc103555371)

[2.2.2 经济可行性 6](#_Toc103555372)

[3 系统设计 6](#_Toc103555373)

[3.1 系统概要设计 7](#_Toc103555374)

[3.1.1 物理架构 7](#_Toc103555375)

[3.1.2 逻辑架构 7](#_Toc103555376)

[3.1.3 分层架构 8](#_Toc103555377)

[3.2 硬件层设计 8](#_Toc103555378)

[3.2.1 电机 8](#_Toc103555379)

[3.2.2 显示屏 9](#_Toc103555380)

[3.2.3 串口 9](#_Toc103555381)

[3.2.4 硬件控制器 9](#_Toc103555382)

[3.2.5 摄像头 9](#_Toc103555383)

[3.2.6 激光雷达 9](#_Toc103555384)

[3.2.7 车身主机 10](#_Toc103555385)

[3.3 驱动层设计 10](#_Toc103555386)

[3.3.1 电机驱动 10](#_Toc103555387)

[3.3.2 显示屏驱动 11](#_Toc103555388)

[3.3.3 串口驱动 11](#_Toc103555389)

[3.3.4 硬件控制器驱动 11](#_Toc103555390)

[3.4 操作系统层设计 12](#_Toc103555391)

[3.4.1 FreeRTOS 12](#_Toc103555392)

[3.4.2 ROS 12](#_Toc103555393)

[3.5 应用层设计 13](#_Toc103555394)

[3.5.1 全向移动 13](#_Toc103555395)

[3.5.2 定位与建图 13](#_Toc103555396)

[3.5.3 路径规划与导航 14](#_Toc103555397)

[3.5.4 用户交互界面 14](#_Toc103555398)

[4 系统实现 15](#_Toc103555399)

[4.1 硬件层实现 15](#_Toc103555400)

[4.1.1 电机 15](#_Toc103555401)

[4.1.2 显示屏 15](#_Toc103555402)

[4.1.3 串口 16](#_Toc103555403)

[4.1.4 硬件控制器 16](#_Toc103555404)

[4.1.6 机器人整体外形 17](#_Toc103555405)

[4.2 驱动层实现 17](#_Toc103555406)

[4.2.1 电机驱动 18](#_Toc103555407)

[4.2.2 显示屏驱动 19](#_Toc103555408)

[4.2.3 串口驱动 20](#_Toc103555409)

[4.2.4 硬件控制器驱动 21](#_Toc103555410)

[4.3 操作系统层实现 22](#_Toc103555411)

[4.3.1 FreeRTOS 23](#_Toc103555412)

[4.3.2 ROS 23](#_Toc103555413)

[4.4 应用层实现 24](#_Toc103555414)

[4.4.1 全向移动 24](#_Toc103555415)

[4.4.2 定位与建图 25](#_Toc103555416)

[4.4.3 路径规划与导航 26](#_Toc103555417)

[4.4.4 用户交互界面 29](#_Toc103555418)

[5 系统测试 31](#_Toc103555419)

[5.1 测试目的 31](#_Toc103555420)

[5.2 测试环境 31](#_Toc103555421)

[5.2.1 硬件环境 31](#_Toc103555422)

[5.2.3 软件环境 31](#_Toc103555423)

[5.3 系统测试 32](#_Toc103555424)

[6 总结 33](#_Toc103555425)

[致谢 34](#_Toc103555426)

[参考文献 35](#_Toc103555427)

# 摘 要

进入21世纪后，传统制造业工厂迫切需要新技术带来的效率提升，人们对于高品质生活的追求也不断增加，多学科融合的机器人系统工程在制造流程与生产技术的转型中所占比重越来越大。在这一背景下，自动导引机器人在制造业、服务业、特种行业的研究与开发成为重点。

本文主要从对各行业关于移动机器人的实际应用场景出发，将应用场景拆分为各基本要素进行解构分析得出项目总体需求，依照嵌入式系统开发逻辑搭建设计方案与项目架构，采用基于STM32的MCU开发技术、多进程/多线程编程、图像处理等技术实现项目功能细节，借助规范化标准接口等工具完成对项目的高可用优化，进而得出一种实现既定目标的解决方案。

最后，通过构造一台可全向移动的智能机器人，利用同步定位与地图构建技术实现软件与现实的拟合，依赖路径规划与导航算法落实自动化执行流程，来实现工业生产装备现代化、居民生活水平智能化的目标。

**关键词**：全向移动模型；同步定位与地图构建；路径规划与导航。

# Abstract

After entering the 21st century, traditional manufacturing factories urgently need the efficiency improvement brought by new technologies, and people's pursuit of high-quality life is also increasing. come bigger. In this context, the research and development of automated guided robots in manufacturing, service, and special industries has become the focus.

This paper mainly starts from the actual application scenarios of mobile robots in various industries, and divides the application scenarios into basic elements for deconstruction analysis to obtain the overall project requirements. According to the embedded system development logic, the design scheme and project architecture are constructed. MCU development technology, multi-process/multi-thread programming, image processing and other technologies realize the functional details of the project, and complete the high-availability optimization of the project with the help of tools such as standardized standard interfaces, and then obtain a solution to achieve the established goals.

Finally, by constructing an intelligent robot that can move in all directions, using synchronous positioning and map construction technology to achieve software and reality fitting, relying on path planning and navigation algorithms to implement automated execution processes, to achieve the modernization of industrial production equipment and the living standards of residents Smart target.

**Key words:** omnidirectional mobility model; simultaneous positioning and map construction; path planning and navigation.

# 绪论

## 1.1 选题背景及意义

随着时代的发展，越来越多的设备朝着现代化、智能化的趋势发展，为了适应现代生产发展的需求，各国相继提出了自己的发展策略。例如德国在2013年汉诺威工业博览会提出的工业4.0概念，现在已经成为了德国的国家战略，当地企业们也展开了一系列生产实验[1]。中国在2014年，由工业和信息化部牵头，会同国家发改委、科技部、财政部、质检总局、工程院等20多个国务院有关部门，组织50多名中国工程院院士、100多位专家起草了《中国制造2025》概念，成为近年来国内制造业的发展纲要[2]。中国大陆是目前全世界最大的工业品出口地和生产地，虽然“中国制造”的标签在国际市场上占据了很大一部分空间，但中国大陆的制造业一直以来都严重依赖国外的设备和技术，这些产业没有完整的自主化体系。为了克服当前国内先进制造业的窘境，我们迫切的需要新技术带来的发展突破。

随着计算机科学和机器人技术的发展，机器人收到了广泛的关注。移动机器人作为机器人的一个分支领域，广泛应用于工业运输、物流、移动作业等诸多领域[3]。自动导引车（AGV）是一种利用各种传感器对未知或已知环境进行识别，进而自主或半自主的移动到目的地的载具。agv通产采用电磁巡线、光学（激光雷达、双目摄像头等）完成对工作环境的采集、识别、分析，利用计算机科学技术将现实环境虚拟化，规划一系列自动化任务。在仓储业、制造业、邮局码头、危险场所和特种行业均有agv的参与。物流行业的相关资料显示，在整个产品生产过程中，加工和制造所占的时间比例只有5%，其余时间都用于储存、装卸、等待加工和运输。在美国，生产成本中的直接劳动成本只占10%，且这一比例还在不断下降，但存储、运输所占成本高达40%。因此，目前世界上的工业强国普遍认为，改造物流结构、降低物流成本是提高企业竞争力的关键措施，物流技术正在逐渐现代化，AGV为传统制造业战略转型提供了重要的技术保障，以成为现代企业设备自动化中的重要一环。

自动导引车（AGV）是一种以电池为动力装有非接触导向装置的无人驾驶自动化车辆，它的主要功能表现为能在计算机监控下按路径规划和作业要求使小车较为精确地行走并停靠到指定地点完成一系列作业功能[4]。本课题选用的agv技术路线为基于二维激光雷达的SLAM分析法，一定程度上降低了对工作环境的人工布置要求，例如不需要铺设大量电磁或光学循迹路线，可以在轻量级预处理的未知场景中直接完成对环境的虚拟化，进一步提高设备的适用性。规模化的agv集群不仅可以技术下沉至居家生活、行业应用等场景，提高人民的生活品质，对于提高工业制造生产效率、节能减排更是具有重大意义。

## 1.2 国内外研究现状

SLAM技术是机器人能够在未知区域中自主工作的关键技术，在未知环境下，机器人通过外部传感器感知环境数据，SLAM算法为机器人提供在环境中的位置并且构造虚拟环境地图，并随着机器人的移动逐步构建剩余未知区域。

SLAM系统架构一般可以分为五个模块，包括传感器数据、里程记、后端、建图及回环检测。传感器数据主要用于采集实际环境中的各类型原始数据，包括激光扫描数据、视频图像数据、点云数据等[5]。里程计主要用于机器人在不同运动状态间的相对位置的测量，如特征匹配等算法；后端主要用于消除里程计带来的累计误差，如滤波器、图优化等算法；地图构建主要用于实际地图的虚拟化；回环检测主要用于空间累计误差的消除。目前，SLAM技术主要分为三种，一种是基于2维或3维激光雷达的激光SLAM，一种是基于单目或多目相机的视觉SLAM，一种是多传感器融合的复杂SLAM。

激光SLAM通过不同时刻的激光点云匹配，计算激光雷达相对运动距离和姿态的变换，实现机器人的同步定位与地图构建[6]。激光SLAM采用2D或3D激光雷达（也叫单线或多线激光雷达），2D雷达一般用于室内机器人，如扫地机器人；3D雷达一般使用于无人驾驶领域。激光雷达的出现使得机器人可以完成速度更快、准确率更高、信息更丰富的环境策略。其采集到的数据一般为分散的、拥有准确角度和距离信息的点，被称为点云。通常情况下，激光SLAM依据不同时刻下的两个点云之间的相似度来进行匹配和对比，计算机器人的相对位姿变化来完成机器人的定位。

视觉化SLAM以由摄像机所得到的连续图象数据为基准,通过环境图象原始数据和成像算法模型计算环境和机器人间的关联,并随着摄像机运动以递增式地确定环境图像,从而输出摄像机在环境图像中的正确定位。早期的机器视觉研究主要采用滤波理论,而非线性的误差模型以及庞大的统计数量则形成了它实用落地的主要障碍。近年来，随着具有稀疏性的非线性优化理论（Bundle Adjustment）以及相机技术、计算性能的进步，实时运行的视觉SLAM成为可能。

多数据融合SLAM采用多种传感器协同处理SLAM流程，例如RGBD深度相机、IMU惯性导航传感器、多线激光雷达等。多传感器采集的信息量大，这些信息的使用方式、可靠性都不相同，其关键在于建立合理的多数据融合模型以提高信息的可信度和算法实时性，同时处理多传感器数据一般需要更先进的传感器与更高阶的数学表达，这也是当下SLAM研究的热点领域。

# 2系统分析

## 2.1 应用场景分析

对于制造业、物流行业来说，传统的agv由于布设难度高、设备复用性差等问题使得agv投入成本偏高，企业更倾向于使用传统的人力作业。而随着近年来传感器质量的提高、计算设备成本的降低，让基于复杂算法的自适应性agv用于实际生产成为可能，越来越多的智能工厂已经陆续投入基于SLAM的AGV用于提高作业效率。

对于居家、服务业场景来说，随着人民日益提高的对美好生活的向往，解放人类劳动力的产品逐渐成为人们的需求。市面上已经出现例如扫地机器人，送菜机器人等AGV产品，送餐机器人送餐过程平稳，减少人送餐出现的视觉影响，提升了科技信息化的带入感[7]。

对于特种用途来说，全自动地形勘探、地下洞穴扫描、无人机集群作业等应用场景也不断的随着科学技术水平的提升而出现。

总的来说，基于SLAM的AGV拥有大量应用场景，能够提高多个领域的综合生产力，提高装备的现代化水平。

## 2.2 功能需求分析

### 2.2.1 需求概述

本系统根据对应用场景的分析，从系统涉及的业务、用户、功能角度出发，逐步由抽象到具体，将项目定义明确、细化，总结得出四个顶层需求。

- 全向移动：机器人可以灵活的进行机动

- 定位与建图：机器人可以感知周围环境并记录

- 路径规划与导航：机器人可以自主计算工作路径并执行

- 用户交互界面：机器人需要具备与人进行交互的界面

### 2.2.2 全向移动

根据应用场景分析，工作环境中的目标实体发生空间位置转移是一个较为普遍的活动。在这个活动中，根据不同的生产任务规划，目标实体在转移过程中的初始位置与目标位置具有空间任意性。并且在同一工作环境中，为了提高生产效率，生产主体的生产方式一般为多个目标实体同时进行转移，此时的工作环境并不是严格意义上的连续无障碍空间，目标实体在转移过程中存在与环境中其他实体发生物理碰撞的可能。Mecanum轮是由瑞典MecannumAB公司所涉及的一款全向轮，可实现前后直行、左右横移、45度运动等[8]。所以在目标实体可交互的前提下，为了最大限度保证目标实体在任意时空中的灵活转移，本项目实体的基本需求为基于麦克纳姆轮的全向移动。

### 2.2.3 定位与建图

根据应用场景分析，本项目实体的工作环境一般为已知或未知的区域，为了保证项目实体能够在此环境下完成生产任务，其关键在于项目实体能否准确判断当前所处位置（定位）并分析全局区域信息构建虚拟地图（建图）。具体来说，项目实体可以从未知环境中的未知地点出发，在运动过程中通过不断刷新传感器所探测到的环境特征确定当前的定位，再根据不同扫描结果的对比从而不断扩大虚拟地图，从而为完成生产任务创造基本条件，这种技术的全程为同步定位与地图构建（Simultaneous localization and mapping, SLAM），是本项目的核心需求。

### 2.2.4 路径规划与导航

根据应用场景分析，本项目实体在已知当前位姿与环境信息之后，还应该具有主动规划当前位置与目标位置之间的一条可行路线的功能，这一由算法计算得出的可行路线一般为环境栅格空间的全局最优路径。同时还需要根据环境的具体情况对这条路线进行高阶运动学优化，例如运动轨迹曲线的凸优化，以满足降低实体运行功耗，缩短通过时间，提高系统运行效率等实际需要。并且项目实体可以依据这条路线进行实际的运动位移，通过监测或控制载具的位置、速度等，与目标点进行对比进而引导载具到达指定点，以完成既定生产任务。

### 2.2.5 用户交互界面

根据应用场景分析，本项目在已经具备一定自动化执行流程之后，人类作为任务规划与控制的发起方，项目需要一种科学合理的人机交互界面。交互界面的基本设计准则为：安全、高效、简单、人性，上述原则的详细体现为：当交互中的某一方做出危险操作时界面必须做出相关提示；系统需要了解用户与机器的工作流程，使双方能够便捷快速的达成目标；界面要避免UI设计对执行程序造成不良干扰，尽可能由简单直接的交互逻辑与直观必需的元素组成；考虑到交互界面是由人操作的，所以界面的设计需要符合人体工程学，积极避免反人类的操作流程。

## 2.2 可行性分析

### 2.2.1 技术可行性

该部分主要将从项目开发所涉及的软硬件平台、系统布局和结构、系统相关技术进行分析。

本项目采用的硬件平台主要为amd64架构处理器，arm-v7架构处理器与控制器；软件平台主要为Linux发行版Ubuntu18.04操作系统、嵌入式实时操作系统FreeRTOS，这些平台均有成熟设计方案与悠久的研发历史，其运行速度、运行稳定性、安全性评估等设计标准均满足项目需求，并且市面上已有类似成熟项目运行于这些平台之上，软硬件平台可行。

本项目采用的系统布局与结构，在物理上为主机-从机二元结构，在逻辑上为模块间P2P的松耦合网络连接。这样的物理结构可以将执行机构与用户交互的主体区分开来，良好的兼容了多个用户可以控制多个机器人的特性。而基于ROS的松耦合处理架构则可以提高机器人开发过程中的代码复用率；并且基于以上架构，不同的模块能够被单独设计，运行时只需要满足接口定义就可以实现模块与模块间的协同运行，不仅提高了项目的开发效率，而且使得项目整体结构更加稳定、可靠。系统布局与结构可行。

本项目采用的相关技术主要有C/C++、Python、STM32驱动开发，FreeRTOS操作系统，基于TCP/UDP的C/S与P/S消息通信模型，基于ROS+Gazebo的模型仿真，多进程/多线程编程，基于PYQT的图形化界面设计。以上技术成熟可靠、标准规范、文档丰富，满足项目需求，开发人员对以上技术具有丰富的开发经验，能够在短时间内完成针对项目本身的高可用、可生产代码。系统相关技术可行。

综上所述，本项目技术可行。

### 2.2.2 经济可行性

该部分主要将从本项目的设备成本、预期收益进行分析。

本项目的设备成本主要包括开发设备成本与生产设备成本，开发设备主要包括一台通用主机、Window操作系统，VM Ware Player虚拟机、Pycharm、VScode，以上开发设备均以获得相关许可，可以合法使用。生产设备分为软件部分与硬件部分，软件部分与开发设备的使用许可基本一致，不再具体分析；硬件设备主要有：4个小型麦克纳姆轮、4个直流编码有刷电机、电机驱动版、树莓派4B、广角免驱摄像头、2维机械式激光雷达、动力电池、导线若干、TTL串口转接器若干。以上硬件设备成本在1000元以内，且大部分设备在我校专业实验室均有提供，设备成本总体可控，满足项目完成需求。

本项目的预期收益将从当前市场对AGV的需求情况以及政策层面进行估计分析。据GGII初步统计，2021年中国市场移动机器人合计销量6.11万台（不包含出口海外），同比增长58.17%，主要驱动因素为智能物流及智能制造需求的带动，其中以3C、锂电池、新能源汽车、光伏、半导体等领域的需求较为突出。从政策层面看，2021年12月28日，工业和信息化部会同国家发展改革委、科技部等共十五个部门，联合印发了《“十四五”机器人产业发展规划》，其中明确提到将重点研制AGV、无人叉车，分拣、包装等物流机器人。由以上两点可以得出，项目预期收益较高。

综上所述，本项目经济可行。

# 3 系统设计

## 3.1 系统概要设计

本项目将根据需求分析与可行性分析中提到的关键点，首先使用逻辑架构和物理架构定义项目雏形，之后利用嵌入式系统设计技术，依据项目所涉及的功能将模块进行分层设计，进而得出完善合理的项目系统设计。

### 3.1.1 物理架构

系统的物理架构规定了组成系统的物理元素，这些物理元素之间的关系，以及它们部署到硬件上的策略。本项目逻辑架构如图3-1所示。

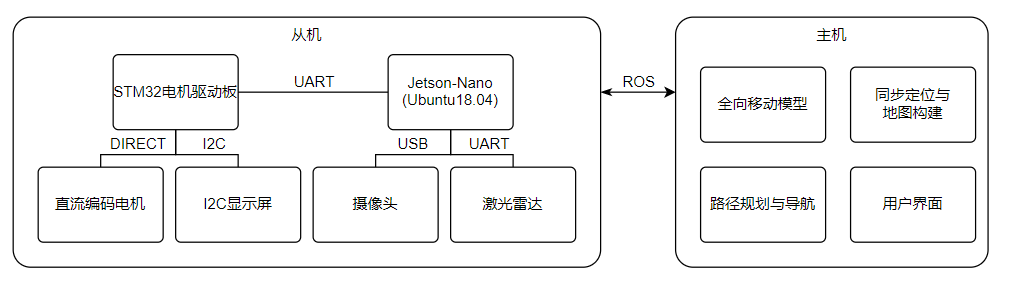


图 3-1 物理架构图

### 3.1.2 逻辑架构

系统的逻辑架构是对整个系统从思想的分类，把系统分成若干个逻辑单元，分别实现自己的功能。本项目逻辑架构如图3-2所示。

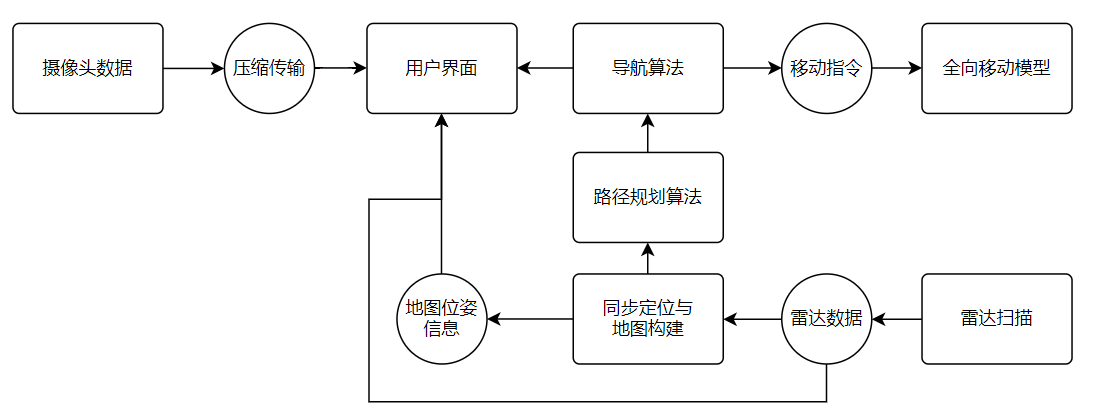


图 3-2 逻辑架构图

### 3.1.3 分层架构

系统的分层架构是根据系统各部分关注点的不同来进行层次间的隔离，以应对不同需求的变化，使得这种变化可以独立进行，降低了技术移植和项目开发的复杂度。本项目分层架构如图3-3所示。

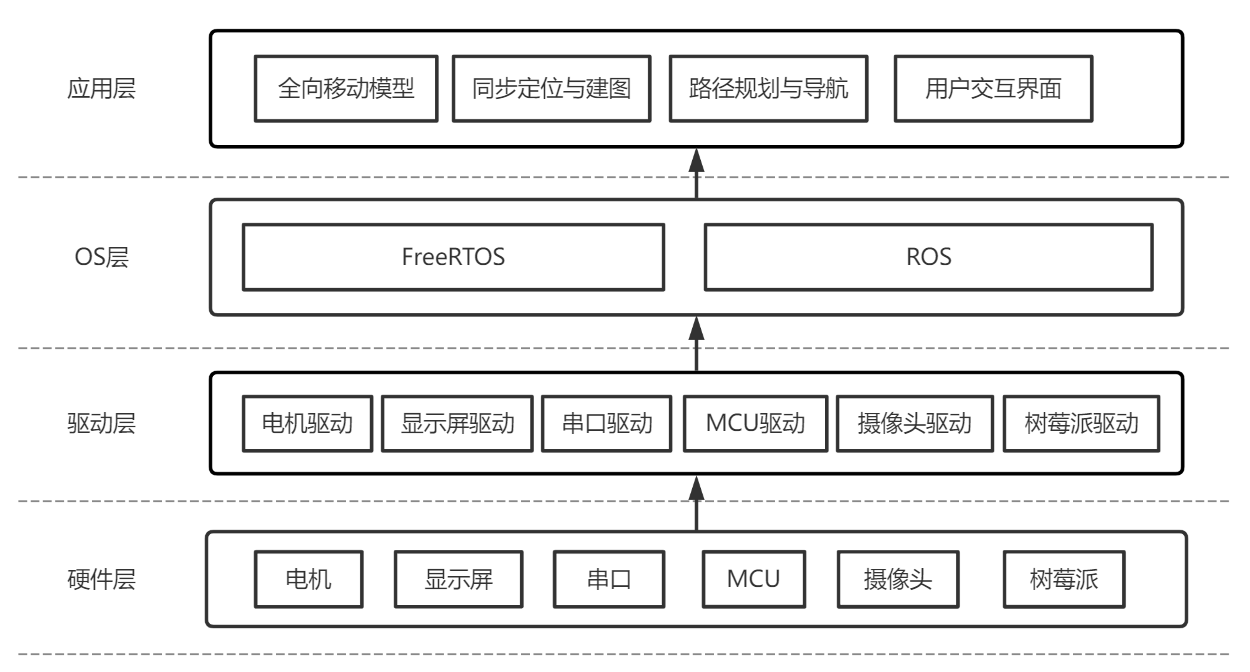


图3-3 分层架构图

## 3.2 硬件层设计

本部分主要介绍电机、显示屏、串口、硬件控制器、摄像头、激光雷达、车身主机的硬件层设计。

### 3.2.1 电机

电机是传动及控制系统中的重要组成部分，根据电机的转动方式不同，一般可以分为：有刷电机、无刷电机、步进电机。在本项目中电机的主要作用是驱动整车运动，需要支持转速不高、负载能力较大、较为精确的转动控制的使用场景，故选择直流编码有刷电机与来驱动麦克纳姆轮。

### 3.2.2 显示屏

显示屏是一种输出装置，用于显示影像及色彩，根据显示屏的显象原理可以分为CRT显示屏与LCD显示屏。在本项目中显示屏的主要作用是显示车体运行电机参数以及上层系统需要显示的数据（如ip地址），由于需要显示的信息量不大，且整车车体空间有限，所以选用n寸OLED显示屏用以显示相关信息。

### 3.2.3 串口

串行接口简称串口，是采用串行通信方式的扩展接口，是目前较为普遍的数据传输方式。在本项目中串口的主要作用是车身主机与ECU之间的数据传输。其通信线路较为简单，只要一对传输线就可以实现双向通信，同时车身主机接口种类有限，故采用基于TTL的串口模块来完成车身域的数据通信。

### 3.2.4 硬件控制器

硬件控制器在小型控制系统中根据实现编写好的控制规律，对传感器传入的信息进行处理，生成控制量，控制量再传给执行器。STM32是ST设计的一系列以ARM-Cortex-M为核心的32位微控制器，拥有包括PWM、IIC、SPI等众多外设，集成度高、型号丰富，满足项目需求。所以选用STM32F103系列作为项目的硬件控制器。

### 3.2.5 摄像头

摄像头的工作原理是，图像信息在通过镜头后由感官元件捕获，经过一系列电路及控制组件对这些信息处理并转换为数字信号，并经过串口或并口输入到后端处理器进行图像的还原，在本项目中摄像头的主要作用是观察车体前端的行进情况。

### 3.2.6 激光雷达

激光雷达是以发射激光束探测目标的位置、速度等特征量的雷达系统。在本项目中激光雷达的主要作用为搜集近似二维平面内的障碍物信息，为基于激光雷达的SLAM算法提供可用信息，激光雷达根据探测范围的不同一般可以分为单线和多线激光雷达，其中多线激光雷达一般用于3维地形重构，而本项目只需要扫描2维地形即可，所以采用机械式单线激光雷达。

### 3.2.7 车身主机

车身主机的定位类似整车系统中的“大脑”，在本项目中车机的主要作用是接收大容量数据设备的信息、与上位机进行通信并将操作指令发送至下位机。树莓派是只有信用卡大小的微型主机，采用ARM架构处理器，根据版本的不同内存大小为2GB到8GB之间，以TF卡作为系统运行的存储介质，内置多种外设，兼容多种操作系统。该设备符合项目需求，故将其作为车身主机。

## 3.3 驱动层设计

本部分主要介绍电机、显示屏、串口、硬件控制器的驱动层设计，其他驱动程序由厂家提供。

### 3.3.1 电机驱动

在本模块中，驱动方式主要为脉冲宽度调制技术作为电机调速方式，并使用增量式PI控制器实现闭环控制。

脉冲宽度调制（PWM），简称脉宽调制，是模拟信号转换为数字信号的一种技术，其有两个关键概念：频率与占空比。PWM频率的定义为单位时间内完成周期性变化的次数；占空比的定义为在一个脉冲循环内，通电时间相对于总时间所占的比例。

数字信号在线性功率元件（如电机）的能量传递原理可以用面积等效原理来解释，即在PWM频率足够大时，占空比可以等效于电压或功率的百分比表示，以此来描述有刷电机的转动功率。

只靠单方面的输入功率无法定义整车电机的运动模型，对于这一简单而单一的电控模块，需要加入一种闭环控制理论来使电机转动对抗外部干扰。闭环控制系统是指被控对象的被控制量会返送来回影响控制器的输出，在工程实际中，应用最为广泛的调节器控制规律为比例、积分、微分控制，简称PID控制。

### 3.3.2 显示屏驱动

在本模块中，驱动方式主要为将字库中的像素位置信息通过i2c通信协议写入到OLED显示屏中，即可完成相关信息的显示。

i2c是一种串行通信总线，使用多主从架构，由飞利浦公司在1980年代为了让主板、嵌入式系统或手机用以连接低速周边设备而发展。i2c采用双向二线制同步通信数据，分别为时钟线（SCL）和数据线（SDA），主要通过在时钟驱动下，拉高或拉低SDA用以表示数据。

### 3.3.3 串口驱动

在本项目中，车机与控制器采用串口作为其基本的通信方式。由于传输的信息具有一定复杂性，所以在通信中增加了一定的协议，通信协议定义了数据单元使用的格式，信息单元应该包含的信息与含义，连接方式，信息发送和接收的时序，从而保通信中数据顺利地传送到确定的地方。

且在通信中，一般的通信数据的长度是可以支持不定长度的，所以在本项目中将使用STM32中的IDLE中断机制来设计不定长数据的接收。IDLE中断在串口收到一帧数据后发生，而RXNE中断接收到一个字节就会触发，通过这两个中断机制来设计数据收发规则。

### 3.3.4 硬件控制器驱动

在本项目中，作为硬件控制器的STM32拥有多种驱动方式：STM32Snippets、Standard Peripheral Library、STM32Cube LL、STM32Cube HAL。STM32Snippets直翻为“代码片段”、“裁剪”，这种开发方式在国内一般被叫做“寄存器”开发，寄存器开发指的是基于ST公司发布的一系列高度优化后的代码片段，使用符合CMSIS标准的直接寄存器读写来减少代码开销，这种开发经常被底层开发人员使用，讲究如何以最小的内存占用使用STM32外设。

Standard Peripheral Librady，即更为常见的标准库开发，每次针对不同型号的STM32都需要手动配置寄存器，这导致了程序维护困难，代码在不同MCU之间的移植性不高。

LL与HAL则是在基于标准库的更上一层的封装，其中HAL是Hardware Abstraction Layer的缩写，直译为硬件抽象层。在这个层次，硬件的实现被封装为一系列标准接口，开发时不再需要关注所用的MCU型号，极大的提高了开发效率，同时还有多种基于HAL的第三方中间件，如RTOS、USB、TCP/IP与图形界面等。

综合考虑三种驱动方式的优缺点，最终将HAL作为硬件控制器STM32的驱动方式。

## 3.4 操作系统层设计

本部分主要介绍FreeRTOS、ROS的操作系统层设计。

### 3.4.1 FreeRTOS

为了充分利用微控制器的资源，以及运行各种硬件驱动的需要，微控制器上搭载了FreeRTOS。FreeRTOS的设计小巧且精简，内核只有3到4个c文件，大部分代码都是以c语言编写，便于阅读、移植和维护。拥有多线程、多作业、互斥锁、信号量和软件计时器等功能。

在该项目中主要制定了三个任务，即电机及运动控制、OLED显示屏驱动、不定长串口数据接收。

### 3.4.2 ROS

机器人是一种高度复杂的系统性工程，机器人设计包含了机械加工、机械结构设计、硬件设计、嵌入式软件设计、上层软件设计等，是各种硬件与软件的集成，甚至可以说机器人系统是当今工业体系的集大成者。

ROS是一套机器人通用软件架构，提高了机器人开发效率。ROS不是传统意义上的操作系统，它异构计算机群上的主操作系统提供一个结构化的通讯层，是一个机器人软件平台[9]。其具有以下特点；进程分布式框架，可使进程分布于不同主机从而分散算力；松耦合，ROS中的功能模块封装于独立的功能包或抽象功能包，包内的模块以节点为单位运行，使用ROS标准IO接口，开发者不需要关注模块内部实现即可完成项目搭建。节点间的通信采用双ROS之间的通信机制来实现PC与工控机之间通信，该方法只需简单的配置即可实现通信[10]。

## 3.5 应用层设计

本部分主要介绍全向移动、定位与建图、路径规划与导航、用户交互界面的应用层设计。

### 3.5.1 全向移动

轮式移动机器人的运动模型可分为完整约束和非完整约束，完整约束的定义为可以用一个由位形变量x，y，θ组成的方程来描述，例如全向移动模型；非完整约束的定义为只能用位形变量的微分方程描述，无法积分成一个位形变量的约束方程，例如双轮自行车模型和双轮差速模型。双轮差速模型机械结构简单、仅使用两个电机驱动，运动控制模型简洁，但旋转运动性能偏弱、狭窄区域通过性差。全向移动模型机械结构复杂，运动控制模型复杂，但可以任意运行于狭窄空间中，灵活完成各种任务，灵活性远高于传统移动平台。

对比研究后，本项目选用基于四轮麦克纳姆轮实现的全向移动模型作为其移动方式。

### 3.5.2 定位与建图

SLAM技术要求实体拥有某型传感器用来收集现实场景的数字化信息，较为普遍的为激光雷达或摄像头，其中激光雷达的传感数据较为精确，且计算量较小；摄像头的数据较为粗略，且计算量较大。不同的传感器需要搭配不同的地图构建算法.此外，还需要考虑每个算法的中央处理器 (CPU) 负载和内存使用情况[11]。Hector-SLAM算法仅需要2维激光雷达的传感数据即可运行，降低了项目所需硬件种类，但其本身要求雷达传感器较为精确。

对比研究后，本项目选用基于二维激光雷达的Hector-SLAM作为其定位与建图方式。

### 3.5.3 路径规划与导航

路径规划算法根据规划模式的不同，分为基于搜索与基于采样的算法，前者例如Dijkstra算法、BFS算法、A-Star算法、D-Star算法；后者例如快速随机搜索树（RRT）算法、快速行进树（FMT）算法、Batch-Informed树（BIT）算法。基于搜索的路径规划算法发展较为成熟并且以及得到了广泛的应用，如游戏人物与移动机器人的路径规划。而基于采样的路径规划算法不需要显式构建整个配置空间和边界，并且在高维度的规划问题中得到广泛应用。A-Star在Dijkstra算法的基础上增加了启发式特性，搜索的效率大大提升。

轨迹优化算法较多，基于曲线平滑的轨迹优化算法有多项式曲线优化、贝塞尔曲线优化等。多项式曲线优化的优点为：曲率连续、代码实现较为简单，缺点为无法固定起点与终点的曲率，无法控制最大曲率，需要控制多项式次数来调整曲线，在路径规划中无法控制优化后路径是否与障碍物碰撞。而贝塞尔曲线具有的凸包性质则可以避免路径与障碍物的重合。

对比研究后，本项目选用A-Star算法与贝塞尔曲线优化作为其导航方式。

### 3.5.4 用户交互界面

交互软件可以实时准确地读取下位机的各种信息，并以一定方式显示在计算机屏幕上。对于操作人员来说，只需要操控计算机即可实现对重载 AGV 的实时监控[12]。移动机器人平台软件系统设计的好坏关系到移动机器人能否正常实现其功能，以及运行的稳定性、实时性和鲁棒性等[13]。

针对此类机器人，市场上较为普遍的交互方式是运行于操作系统平台的图形化界面。该界面所展示的内容包括但不限于字符串、交互按钮、图形显示；界面的运行速度需在人类生理上可接受的范围内；界面的布局不仅需要符合人体工程学而且鉴于项目的特殊性，特殊的动作执行按钮必须放置于醒目、易于操作的位置。

用于交互的图形化界面一般可以是Web界面或应用程序，但本项目的图形化界面需要与操作系统共享资源，例如串口、I/O系统等，并且出于安全性考虑，该界面只能运行于经过安装步骤之后的操作系统之上，而不可以运行于任意浏览器中，所以项目的图形化界面选用跨平台的C++桌面应用程序开发框架qt。同时为了避免界面在同时接收大量数据（如图像、雷达数据）后造成操作卡顿，界面的运行时环境设计为基于多线程的异步处理逻辑。

对比研究后，本项目选用基于PYQT5的多线程GUI界面作为其交互方式。

# 4 系统实现

## 4.1 硬件层实现

### 4.1.1 电机

电机的硬件层实现主要包括电机驱动芯片的原理图以及实际接线方式，如图4-1与图4-2所示。

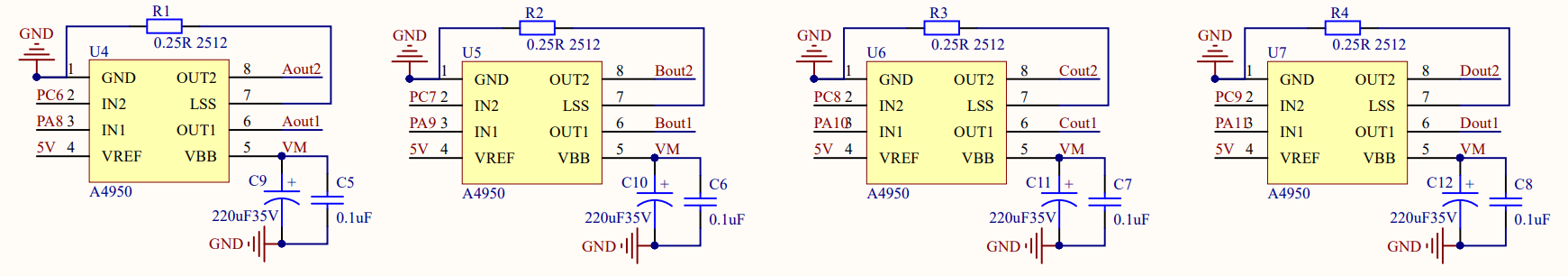


图 4-1 驱动芯片原理图

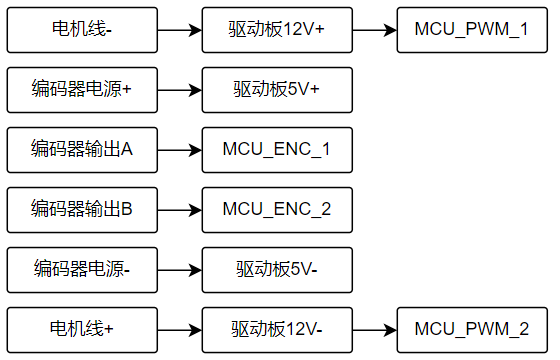


图 4-2 电机接线

### 4.1.2 显示屏

显示屏的硬件层实现主要包括实际接线方式，如图4-3所示。

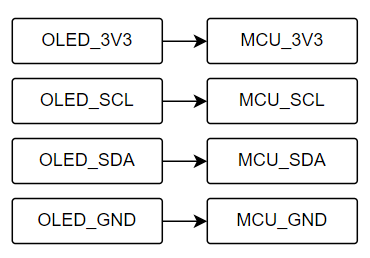


图 4-3 显示屏接线

### 4.1.3 串口

串口的硬件层实现主要包括实际接线方式，如图4-4所示。

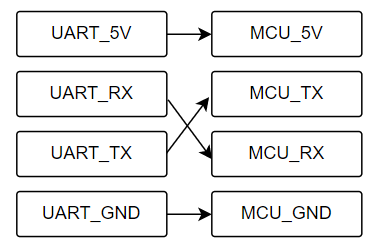


图 4-4 串口接线

### 4.1.4 硬件控制器

硬件控制器的硬件层实现主要包括原理图，如图4-5所示。

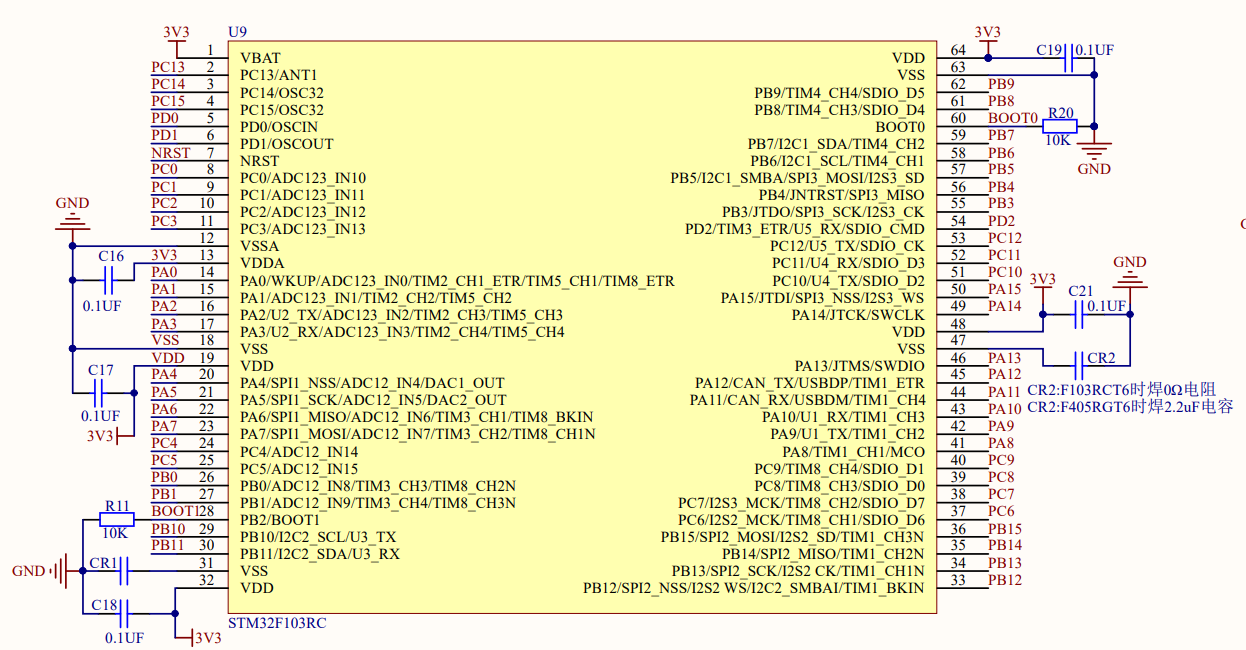


图 4-5 MCU原理图

### 4.1.6 机器人整体外形

机器人整体外形如图4-6所示。

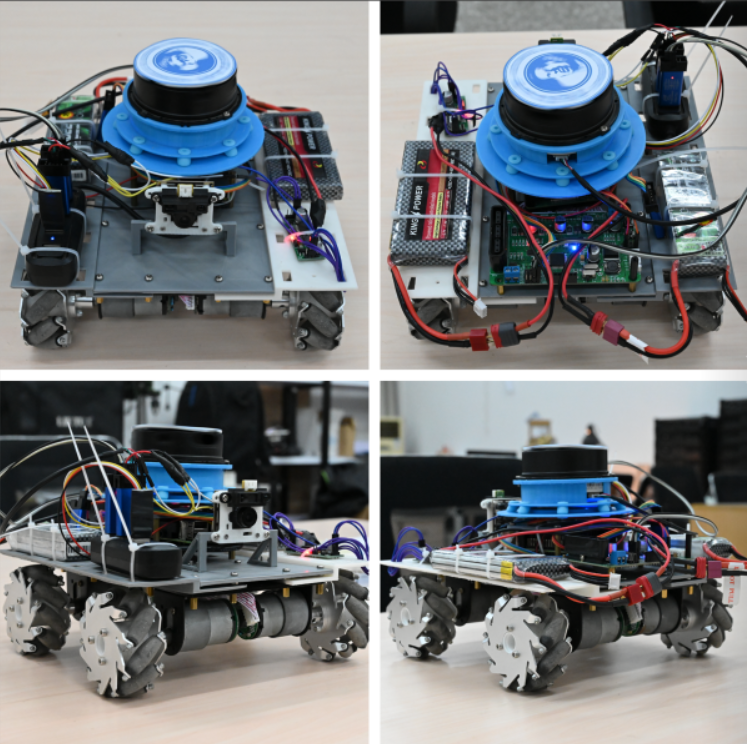


图 4-6 机器人整体外形

## 4.2 驱动层实现

### 4.2.1 电机驱动

电机的驱动层实现主要包括MCU相关配置、PID控制算法，如图4-7与4-8所示。在现代随机控制以及随机信号处理技术中，存在着大量的随机时变的噪声[14]，所以电机编码器接收设置为10倍滤波。

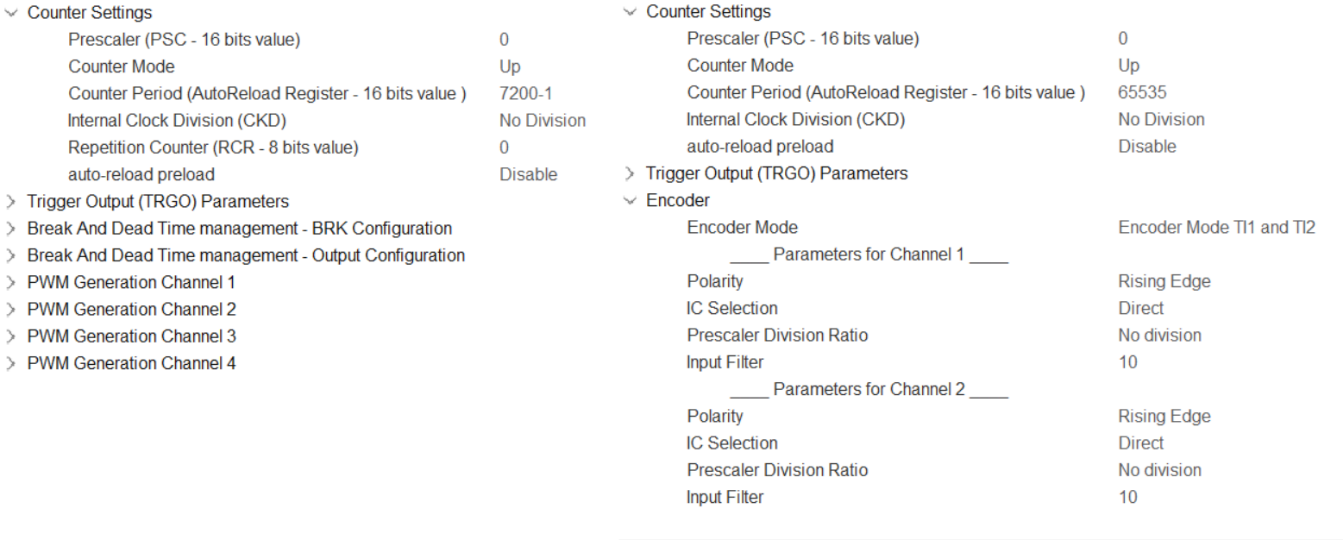


图 4-7 电机MCU配置

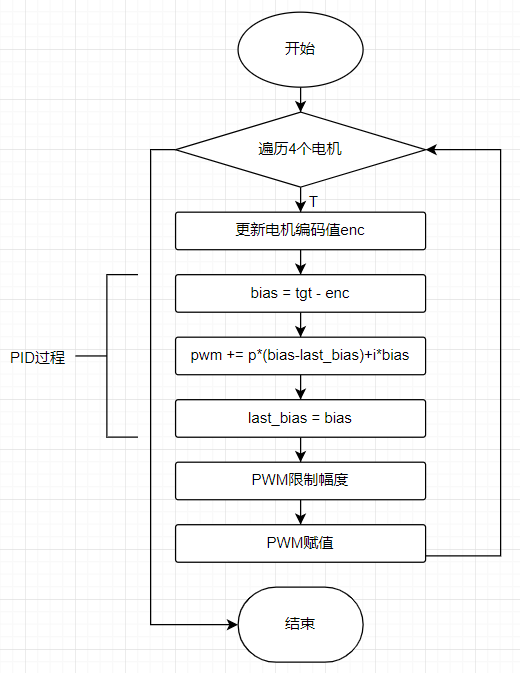


图 4-8 PID控制逻辑流程图

### 4.2.2 显示屏驱动

显示屏的驱动层实现主要包括MCU相关配置、I2C驱动接口，如图4-9与图4-10所示。

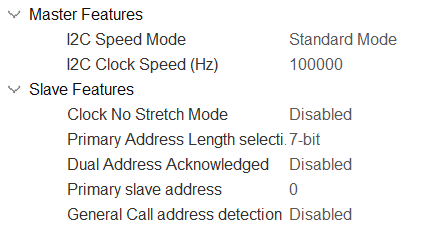


图 4-9 MCU相关配置

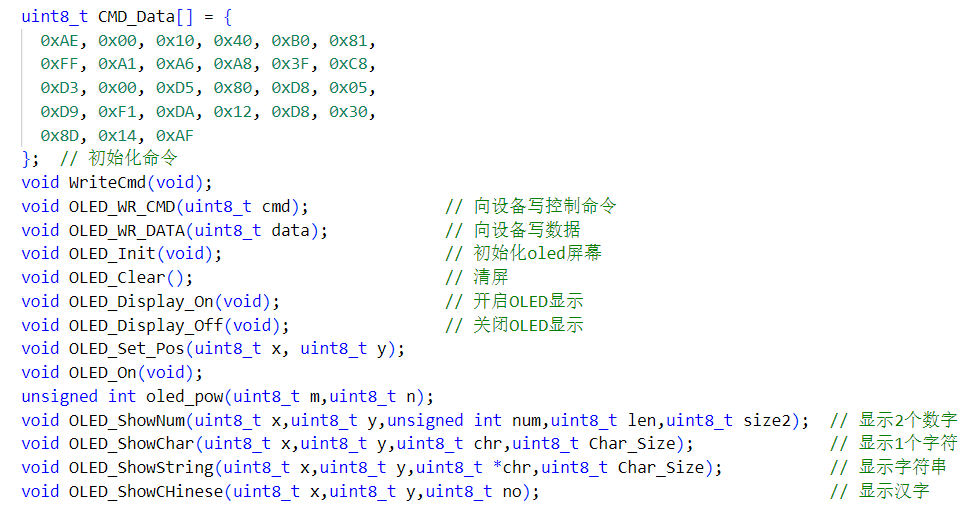


图 4-10 I2c驱动接口列表

### 4.2.3 串口驱动

串口的驱动层实现主要包括MCU相关配置，接收处理逻辑，如图4-11与图4-12所示。

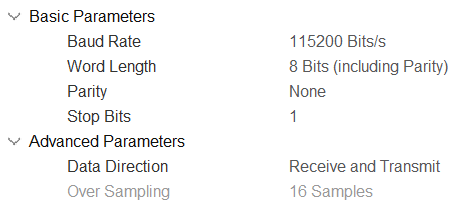


图 4-11 MCU相关配置

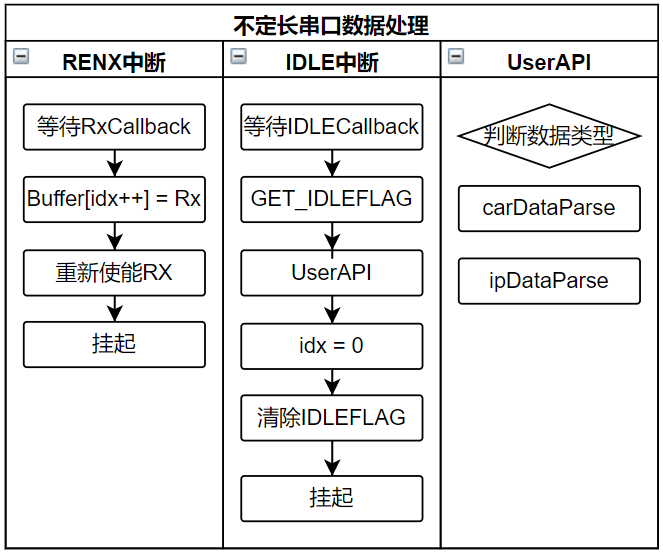


图 4-12 接收处理逻辑

### 4.2.4 硬件控制器驱动

硬件控制器的驱动层实现主要包括MCU接线引出，系统配置，如图4-13与图4-14所示。

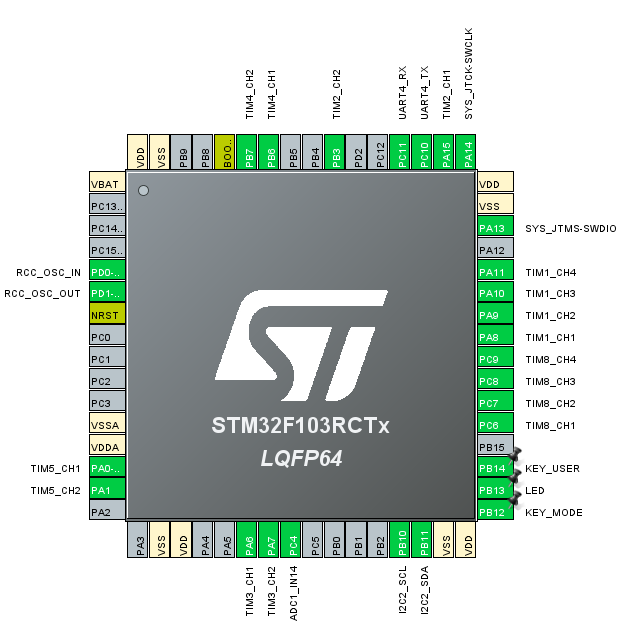


图 4-13 MCU接线引出

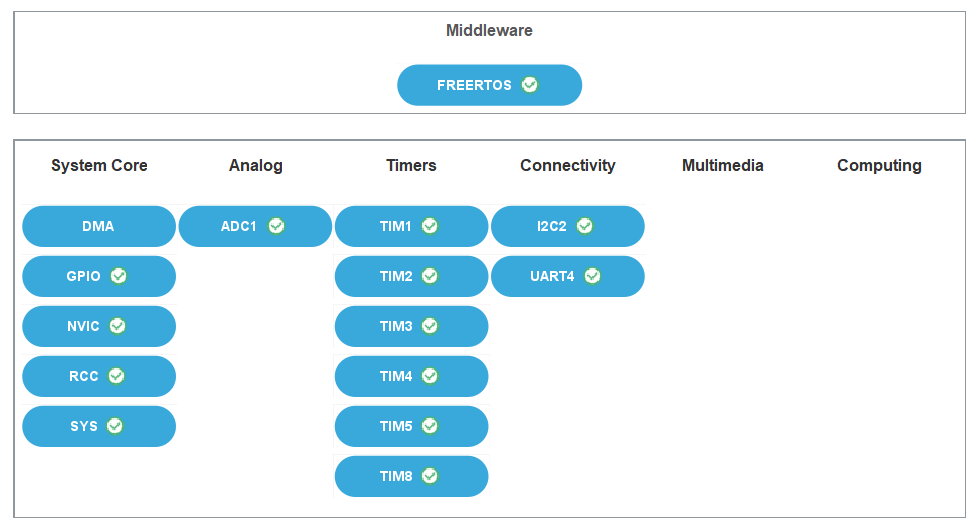


图 4-14 MCU系统配置

## 4.3 操作系统层实现

### 4.3.1 FreeRTOS

FreeRTOS的操作系统层实现主要包括OS系统配置，线程表，如图4-15与图4-16所示。

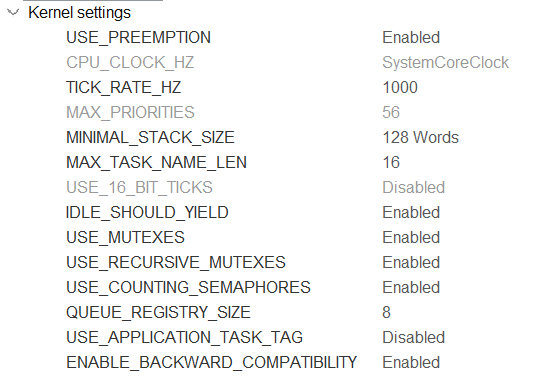


图 4-15 OS系统配置

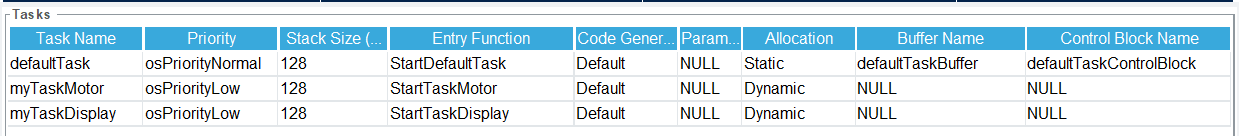


图 4-16 线程表

### 4.3.2 ROS

ROS的操纵系统层实现主要包括ROS节点部署，如图4-17所示。

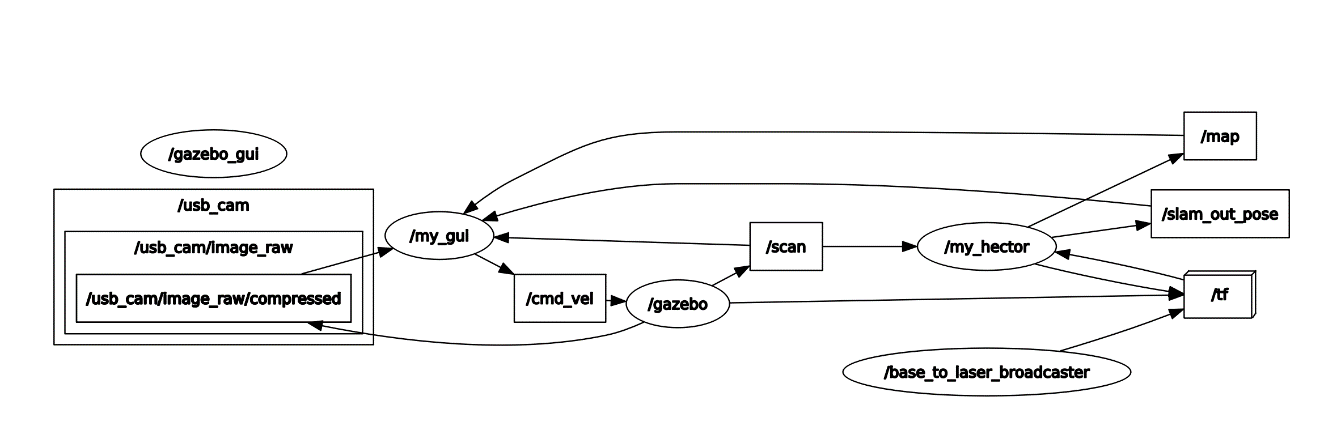


图 4-17 ROS节点部署

## 4.4 应用层实现

### 4.4.1 全向移动

麦轮的安装方式根据系统设计得出的方案以O-长方形进行安装，如图4-18所示。

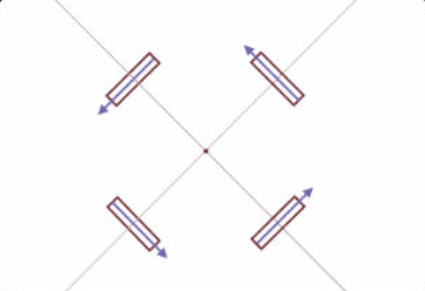


图 4-18 麦轮安装方式

运动学建模方法一般分为两种，正运动学模型指的是由轮子的运动状态推导出底盘的运动状态，逆运动学模型指的是由底盘的运动状态堆导出轮子的运动状态。在这里，底盘的运动可以用三个变量来表示：x，y，z，分别表示底盘沿x方向的速度、沿y方向的速度、旋转的角速度，如图4-19所示。

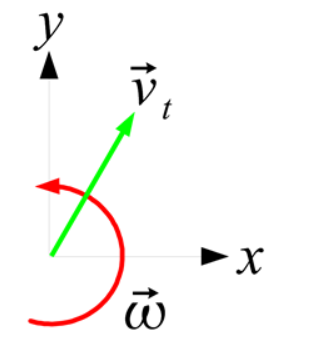


图 4-19 xyz速度表示

该系统作为一个纯线性的刚体运动系统，可以通过观察底盘实际运行方式，将底盘运动状态简单相加从而完成逆运动学模型的构建。

当底盘沿着x轴平移时：

当底盘沿着y轴平移时：

当底盘绕几何中心自转时：

将三个方程组相加，即可得到最终的麦轮底盘的全向移动运动学反解。

### 4.4.2 定位与建图

Hector-SLAM的算法本质很简单，即将现在激光雷达扫描到的点云信息与已知的点云信息做匹配，并将已经匹配到的信息进行扩展，直到探索完毕，其运行的基本流程如图4-20所示。

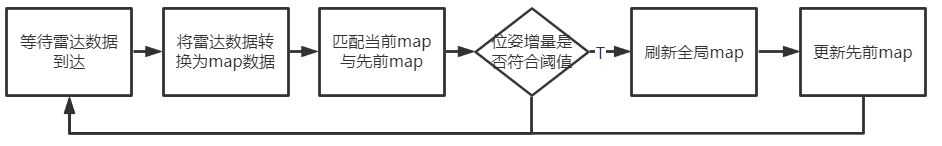


图 4-20 hector算法流程

其中hector算法的核心为“扫描匹配”，扫描匹配是将激光雷达扫描与现有地图对齐的过程，方法是基于带有地图记忆的端点对齐的优化方法，Hector-SLAM扫描匹配的基本思路是使用高斯-牛顿法求得当前的位姿，从而得出当前机器人最佳位姿。

其核心公式为[15]：

(1) : 机器人在世界坐标系二维平面的位姿

(2) 和: 世界坐标系二维横纵坐标

(3) : 相对于世界坐标系y轴的旋转角度

(4) : 上一时刻机器人的位姿

(5) : 本次最优位姿，即扫描匹配目标的结果

(6) : 激光坐标系转换到世界坐标系

根据公式推导可以得出：

再通过双线性插值法将地图信息表示到栅格地图中：

至此我们就完成了利用栅格地图表示的hector地图构建功能。

### 4.4.3 路径规划与导航

a-star算法综合了广度优先算法和Dijkstra算法的优点，在保证可找到最优路径的同时依赖辅助函数进行启发式的搜索增加算法效率。

其算法流程如图4-21所示：

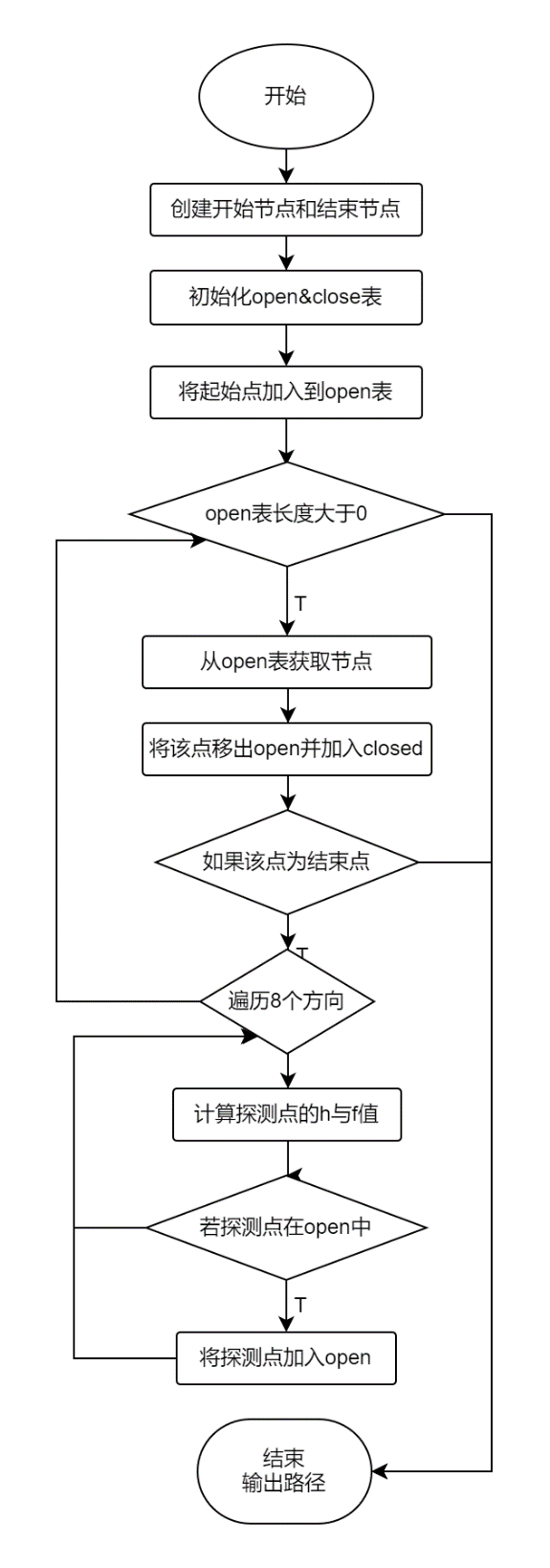


图 4-21 a-star算法流程

其中，启发函数的选取决定了算法的执行效率与是否能找到全局最优路径，启发函数越是没有“启发性”，那么算法找到的路径将更加接近全局最优解；而启发函数“启发性”越高，算法运行效率也就越高，但有可能不是全局最优解。启发函数的一般表达形式为`f(n)=g(n)+h(n)`，g表示从起点到n点的实际距离，h表示n点到目标点的估算距离，其拥有以下特性：

(1) g(n)=0: 算法转换为使用贪心策略的广度优先算法，速度最快，但可能不是最优解。

(2) h(n)小于实际距离: 算法转化为无目标的广度优先算法，可找到全局最优路径但算法效率偏低。

本项目选用起始点与目标点的欧几里得距离作为评估算法距离的启发函数，并且为了提高计算效率，算法所计算的点并不是栅格地图的实际点，而是经过一定缩放的单元格，这样得出的最短路径过于“平整”，如图4-22所示。所以需要曲线优化算法优化车体的实际运行路径。

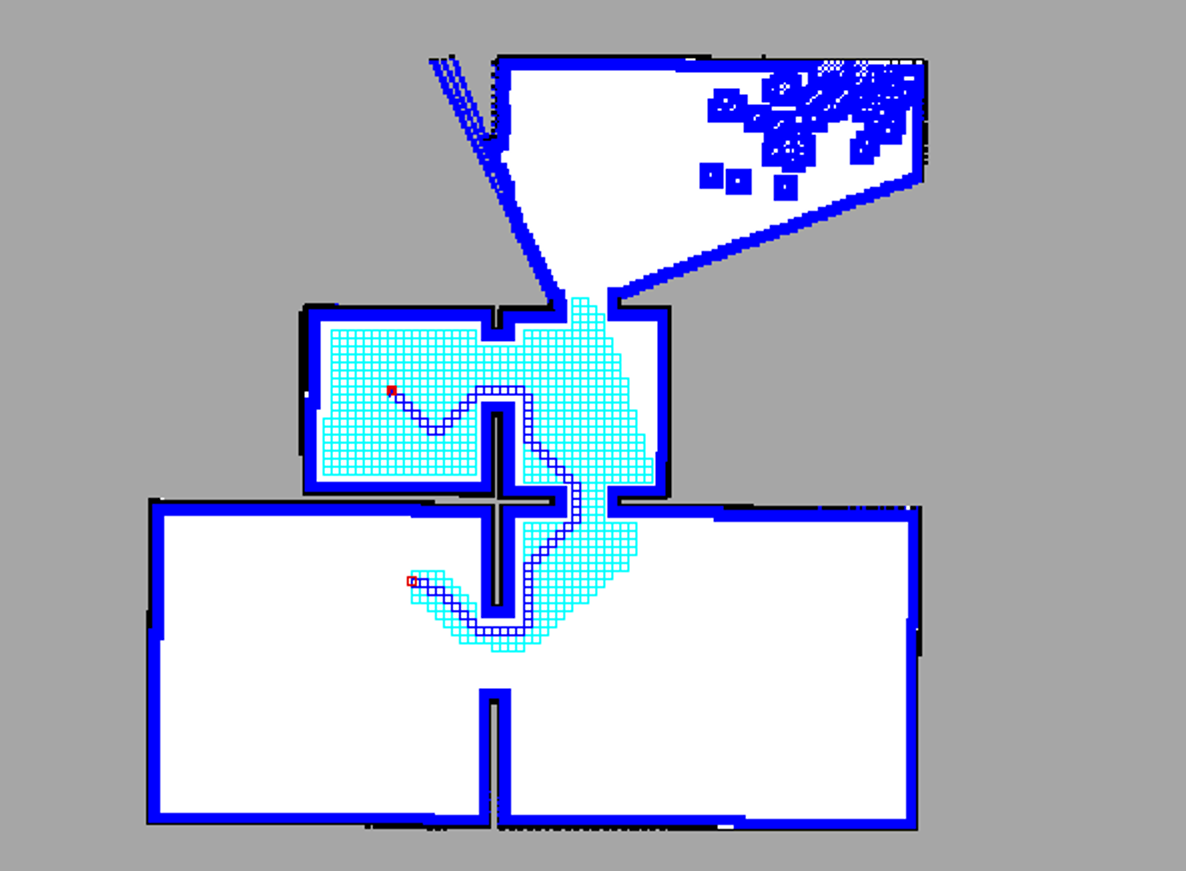


图 4-22 未经轨迹优化的路径

轨迹的优化是一个凸优化问题，在本项目中采用贝塞尔曲线对轨迹进行优化，经过优化的路径如图4-23所示。

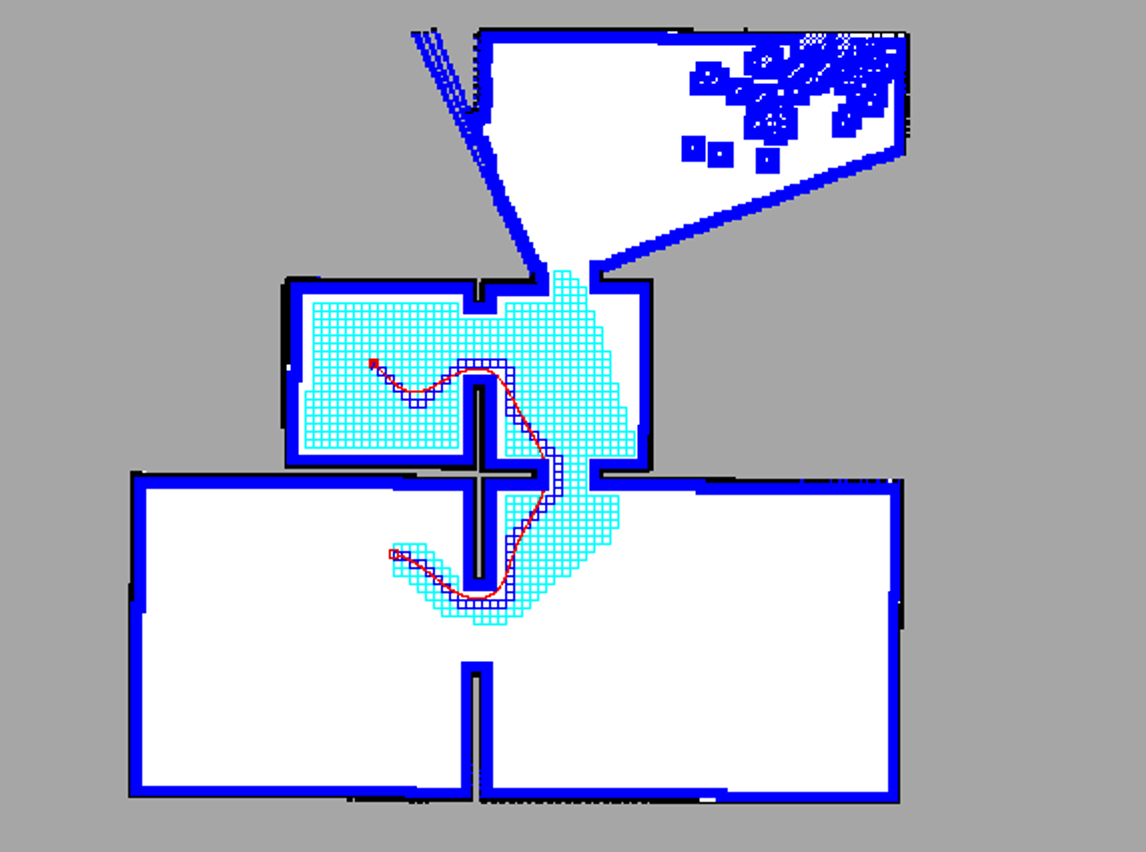


图 4-23 经轨迹优化的路径

### 4.4.4 用户交互界面

界面结构主要有运行库、算法、设计模板、子模块测试部分组成，其基本结构如图4-24所示：

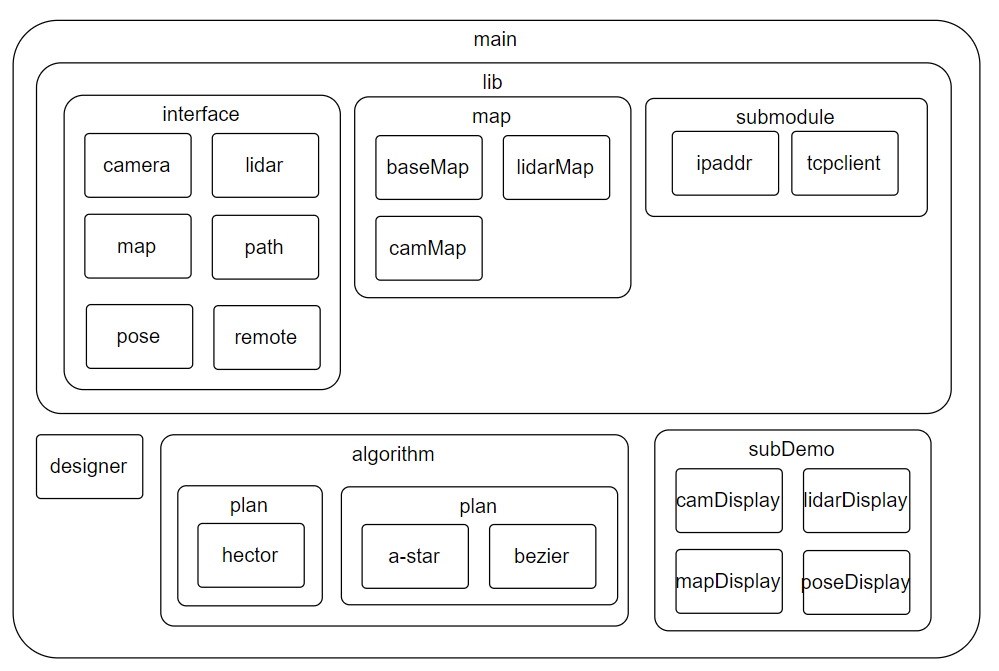


图 4-24 界面设计结构

主要流程以多线程的方式运行，如图4-25所示：

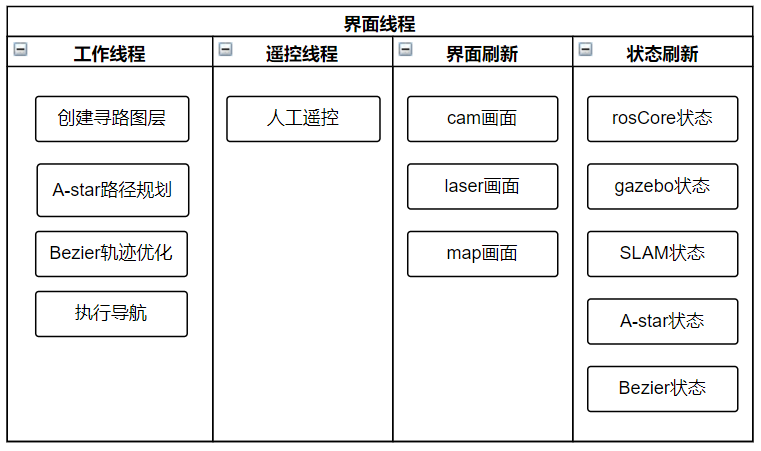


图 4-25 多线程逻辑

最终的界面效果如图4-26所示：

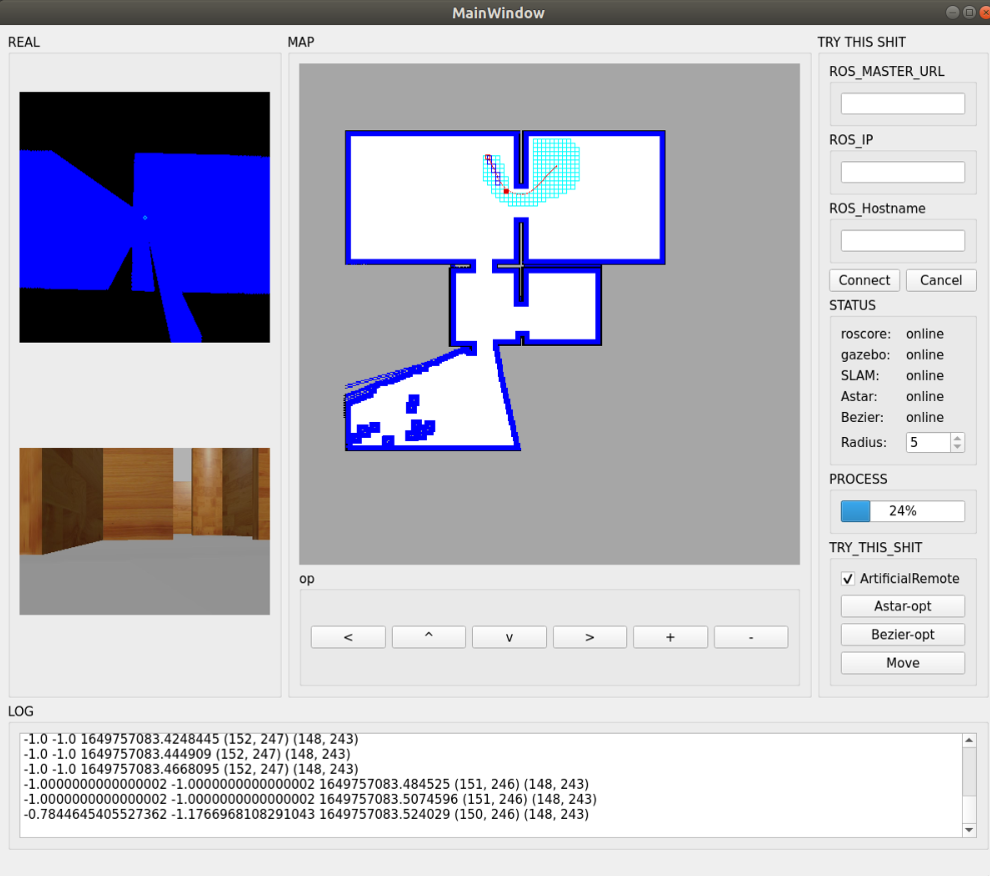


图 4-26 界面显示

# 5 系统测试

## 5.1 测试目的

软件测试一般用于鉴定软件运行的正确性、完整性、安全性和质量，依照可计算理论，一个简单的数学证明推断出下列结果：不可能完全解决所谓的“死机”，换句话说，软件测试是一种实际输出与预期输出间的审核和比较过程。

在测试过程中可以分析比较程序运行的状态，避免系统在交付后出现任何未知的运行错误，降低整体的风险。一个好的测试用例在于它能发现以前未发现的错误，并且为开发过程提供不同维度的信息，让在开发过程中处于不同维度的人做出正确决策。总之软件测试的目的是尽快找出项目中隐藏的漏洞，并及时修复。

## 5.2 测试环境

### 5.2.1 硬件环境

为保证本系统的正确开发和运行，硬件环境应满足以下要求：

- 处理器：Intel i5及以上

- 内存：最低要求8GB，推荐16GB

- 硬盘：50GB以上

- 显示器：无特殊要求的彩色显示器

- 输入设备：键盘与鼠标

### 5.2.3 软件环境

为保证本系统的正确开发和运行，软件环境应满足以下要求：

- 操作系统：Ubuntu18.04

- 编译器/解释器：GNU、Python3.9

- 运行时环境：ROS-melodic

- 驱动程序：激光雷达驱动、串口驱动、摄像头驱动

## 5.3 系统测试

系统测试指的是将项目所需的全部客观事物整合起来一起运行测试，如项目中的软件、硬件、测试场地、测试人员等。其目的在于将最终成品与系统需求分析作比较，发现与系统定义矛盾或不匹配的地方。本次测试主要将从最顶层的需求角度进行测试，分为以下若干测试用例。测试结果如表5-1、表5-2、表5-3所示。

表5-1 同步定位与地图构建

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 01 | 用例名称 | 同步定位与地图构建 |
| 测试环境 | Ubuntu18.04系统、ROS-melodic系统 | | |
| 前提条件 | 软硬件运行环境搭建完毕 | | |
| 测试步骤 | 1. 人工控制机器人沿x轴前进。 2. 人工控制机器人沿y轴前进。 3. 人工控制机器人沿yaw轴旋转。 4. 观察界面地图构建效果。 | | |
| 输入数据 | 键盘控制指令。 | | |
| 预期输出 | 较为完整的地图构建。 | | |
| 实际输出 | 较为完整的地图构建。 | | |
| 问题描述 | 地图构建随用到的雷达数据没有像仿真环境中的雷达数据平滑。 | | |
| 问题修改摘要 | 整理实际车身上激光雷达周围的布线后问题解决。 | | |

表5-2 路径规划

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 02 | 用例名称 | 路径规划 |
| 测试环境 | Ubuntu18.04系统、ROS-melodic系统 | | |
| 前提条件 | 地图构建完毕 | | |
| 测试步骤 | 1. 点击创建寻路模式； 2. 鼠标点击相较当前车身位置的无障碍目标点； 3. 鼠标点击相较当前车身位置的有障碍目标点； 4. 鼠标点击相较当前车身位置的不可达目标点； 5. 观察寻路效果。 | | |
| 输入数据 | 鼠标点击的位置。 | | |
| 预期输出 | 系统均可计算出上述路径。 | | |
| 实际输出 | 系统在计算不可达目标点时系统未响应。 | | |
| 问题描述 | 计算不可达目标点时算法陷入类似死循环的状态，等待一段时间后才能显示结果。 | | |
| 问题修改摘要 | 通过限制计算范围来裁剪没必要计算的目标点，提高系统运行效率。 | | |

表5-3 导航

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例编号 | 03 | 用例名称 | 导航 |
| 测试环境 | Ubuntu18.04系统、ROS-melodic系统 | | |
| 前提条件 | 路径规划完毕 | | |
| 测试步骤 | 鼠标点击开始执行。 | | |
| 输入数据 | 鼠标点击时的开始信号。 | | |
| 预期输出 | 车身按照既定路线移动。 | | |
| 实际输出 | 车身偶尔会偏离路线。 | | |
| 问题描述 | 在网络状态不佳的前提下，车身在执行大角度转弯时会偏离路线，随后回到正常路线。 | | |
| 问题修改摘要 | 由于位姿的计算是在远程完成的，车身在网络条件不佳的前提下无法即使获取当前位姿从而短暂偏离路线，通过增加延迟确认机制以及计算实际导航精度来解决。 | | |

# 6 总结

本项目为了达到装备智能化的目的，针对一般工业、居家场景下，分析项目的核心需求，基于嵌入式系统设计方法提出了一种切实可行的设计思路，并实现了一台可在室内执行简单任务的自动导引移动机器人。

本项目实现的功能主要有：基于麦克纳姆轮的全向移动控制，基于Hector-SLAM算法的环境地图构建，基于A-star与凸优化方法的路径规划与导航，基于pyqt的用户交互界面，最终可以实现移动机器人在未知环境中的定点自主移动。本项目内容涵盖机械运动学模型搭建，电机控制单元与微控制器的工作原理，计算机科学与技术相关理论，算法中的复杂数学方法等，是一个多学科交叉的工程实践。对于复杂系统开发来说，为了保证项目的开发效率与降低排错难度，一般会依据一套设计原理或方法论自上向下的描述分析该系统，并逐级分层，以接口的方式描述层级之间的交互。完成本项目时通过基于嵌入式系统设计理论的分析，发现了在嵌入式开发领域中在一定程度上也可以遵循软件的设计思路，即利用系统分层和面向对象的设计方法大量简化了嵌入式设备的研发周期及维护难度；同时初步理解了在汽车领域“软件定义汽车”这句口号的现实依据，与本人实习正在学习的AUTUSAR架构存在一定程度的对比学习机会，为此类后续项目的研究提供了一条成熟合理的技术路线与方法论。

同时项目研究还有不足之处：用ROS逻辑模型统一整个项目架构时会忽略其技术细节。对于实时性很高的场景下，ROS分布式节点的同步是使用基于网络的TCP/UDP协议，在项目的模拟仿真研发过程中一般过于理想化网络传输质量，进而无法排除将两个同步性很高的需求在开发过程中被分配到了两个节点，而且这两个节点还是依赖网络的跨主机通信，最终造成的现实问题就是项目在网络条件不佳的情况下，运行导航时无法及时获取当前位姿，造成导航效率低下。

项目中依然存在可以进一步研究的内容：如Hector-SLAM算法在layer数量不同的情况下的表现，实现更先进的图像传输编解码器如H264或H265等，实现更高效的路径规划与轨迹优化算法等。

通过本次研究，使我加深了对嵌入式系统开发理念的认识，尤其是对于软硬件结合的复杂系统开发来说，不仅需要自上而下的对系统拥有良好的抽象分层，而且涉及到一些关键设计还应主动摆脱思维定势积极深挖技术原理。在未来的研究中，我将不断完善设计理念，提高自身技术水平。

# 致谢

美妙人生的关键在于你能够迷上什么东西，对于我来说，本科四年的学习不仅仅是专业技术的掌握，更多的是关于自我的探索，关于事物本质的探索。

我的成长，离不开黄贤英老师对我耐心的指导。上到工作方向下到比赛题目，无论事情本身有多复杂，老师总是能在我最需要方向的关键时刻给予我果断的选择。虽然她的指导在当时的我看来似乎是无关紧要的，有的执行有的没有执行，但站在今天的角度看，这些抉择的背后都蕴含着极度理性的原因和目的，值得被当时的我不加以任何道德或文化评判的研究和追问。这种精神在某次比赛截止的晚上唤醒了我，那时的我面对成山的资料和绞尽脑汁都无法写出的文档，眼看着时间一点点流逝，但就是无法继续前进。那时的我回想起之前的抉择，我无法接受已经进行到最后一步时突然放弃所带来的负面作用，我无法容忍在我生日当晚做出弃赛的决定，最终在黄老师的精神指导下完成了比赛。老师总是能基于现实主义立场给出解决问题的合理建议，以一种复杂但微妙的方式一步步引领我发现人生的方向。

我的成长，离不开同学朋友们与我激烈的辩论。我的矛盾，有时不仅仅是人与人之间的矛盾，也是关于自我本身的否定。怀疑精神使我将表象看作是一种可怀疑的、有限的“假象”，进而用思维、逻辑去追溯这一切背后的那个实体，并把那个看不见的实体而不是看得见的经验表象当作事物的本质；不同观点、理念的碰撞使我能够将各自的观点拆分为诸多最基本的、无法违背的命题，使我将一切感性的事物看作是认识背后那个事物的中介而非终点，当我的解释能力足够强大到对这些要素以一种独立的姿态进行解构分析时，这就是找到实现目标路径的方法。朋友总是敢于提出最赤裸、最核心的矛盾，以一种对抗的方式使我找到进步的方法。

最终，我将携带大学这段时间的思辨，创造属于我自己的人生价值。

# 参考文献

1. 盛斌, 魏方. 德国工业4.0[J]. 中国海关, 2014(8).[54].
2. 黄群慧, 贺俊. 中国制造业的核心能力、功能定位与发展战略——兼评《中国制造2025》[J]. 中国工业经济, 2015(06):5-17.
3. Zhi L, Xuesong M. Navigation and control system of mobile robot based on ROS[C]//2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2018: 368-372.
4. 王智勇, 姬婧媛. AGV 小车的控制系统的设计与开发[J]. 桂林航天工业高等专科学校学报, 2007, 12(4): 44-46.
5. Filatov A, Filatov A, Krinkin K, et al. 2d slam quality evaluation methods[C]//2017 21st Conference of Open Innovations Association (FRUCT). IEEE, 2017: 120-126.
6. 韦晓琴. 基于激光雷达的 AGV 机器人 SLAM 与定位导航研究[D]. 广州: 华南理工大学, 2019.
7. 张博等. 浅谈基于ROS的送餐服务机器人[J]科学与信息化, 2020(18):176-177
8. 华崴鉴. 基于自主导航的移动机器人控制平台的设计与实现[D]. 南京邮电大学, 2019.
9. 白亮亮,平雪良,陈盛龙,等.分布式移动机器人控制系统设计与实现[J].机械设计与制造,2015(10):180-183.
10. 张华健, 钱钧. 基于 ROS 的远程呈现移动机器人系统设计[J]. 制造业自动化, 2019, 9.
11. Dhaoui, Riadh, and Amine Rahmouni. "Mobile Robot Navigation in Indoor Environments: Comparison of Lidar-Based 2D SLAM Algorithms." International Conference on Design, Simulation, Manufacturing: The Innovation Exchange. Springer, Cham, 2021.
12. 程勇, 陈薇. 基于无线网络的重载 AGV 实时监控系统设计[J]. 电气应用, 2019.
13. 张华健, 钱钧. 基于 ROS 的远程呈现移动机器人系统设计[J]. 制造业自动化, 2019, 9.
14. 赵连娟, 韩硕. 基于 Kalman 滤波的 PID 电机控制系统设计[J]. 轻工科技, 2015 (3): 65-66.
15. Kohlbrecher S, Von Stryk O, Meyer J, et al. A flexible and scalable SLAM system with full 3D motion estimation[C]//2011 IEEE international symposium on safety, security, and rescue robotics. IEEE, 2011: 155-160.