# 简易机器人

# 设计说明书

# SDD103

# V4.0

## 分工说明

| 小组名称 | undefined | |
|---|---|---|
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 16061160 | 陈麒先 | 负责第 1,3.2,5,6.6,7 章的编写并对全文进行校对修改 |
| 15071111 | 刘亮 | 负责第 3.1,6.1,8.2 章的编写并参与第一版文档汇总 |
| 16061005 | 秦枫 | 负责第 4, 6.2,6.3 章的编写并参与第一版文档审查 |
| 16061034 | 孙琦 | 负责第 2, 8.1 章的编写并参与第一版文档审查 |
| 16061063 | 周尚纯 | 负责第 3.1,6.4,6.5 章的编写并参与第一版文档审查 |

## 版本变更历史

| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
|---|---|---|---|---|
| V1.0 | 2019.4.21 | 刘亮等 5 人 | 陈麒先 | 设计说明书第一版 |
| V2.0 | 2019.4.22 | 陈麒先 | 刘亮等 5 人 | 设计说明书第二版 |
| V3.0 | 2019.6.8 | 刘亮等 5 人 | 陈麒先 | 设计说明书第三版 |
| V3.0 | 2019.6.14 | 陈麒先 | 周尚纯 | 设计说明书第四版 |
| | | | | |

# 1. 范围

## 1.1 项目概述

### 1.1.1 开发背景

机器人的研究和应用在很长一段时间内都是科学研究的热点。众所周知，每一次技术的变革都是对人类更进一步的解放，而机器人的诞生和应用也不例外。人们希望机器人能在农业、工业、服务业等行业中逐渐取代一些简单基本的人工，这要求机器人具有不亚于人的专业能力。

在机器人领域，日本、美国和欧洲已经有着牢固的技术基础和广泛的应用范围，而我国近年来工业机器人的生产和制造也取得成果并投入生产。世界各国对机器人的需求总体上呈增长趋势，人们对机器人的期望也向着智能化、技术化方向发展[1]。从技术发展的角度来看，就机器人移动和避障这一主题，除人工势场法、栅格法等传统算法外，更有基于神经网络、可视性二叉树、滚动时域控制等新颖算法的提出。

我们选择机器人嵌入式系统的开发，力图在前人开发经验的基础上自主实现具有避障、路径规划和目标抓取功能的机器人，并希望未来能够不断扩充和优化其性能，甚至实现技术突破。

### 1.1.2 功能和需求

我们项目的核心目标是实现一个购物机器人系统，最主要也是最基本的功能是可依用户指令取物，其中包括主动避开障碍物、最优路径规划和多物品识别等子任务。在此基础上，我们将追求更高的要求，即用户可以通过中文语音向机器人下达命令，也可使用我们提供的网站远程控制机器人启动、查看机器人状态。

我们将对成果机器人实现目标任务的程度进行评估和不断优化。具体包括在复杂情景中的避障行为是否正确、路径选择是否较优、识别检测到目标物是否精准、尝试抓取目标物的次数是否有限、是否总能到达目标物所在的较小范围内、是否能稳定实现有效抓取、能够抓取的目标种类和体量等，对于具体评测手段，

我们将在实践和测试中给出量化指标。

### 1.1.3 用户价值与应用场景

我们认为具有自主移动和目标抓取功能机器人的实现是非常有意义的。这两项功能在现实中已经分别有广泛的应用，例如家用扫地机器人、工厂零件装配机器人等。集二者于一体的机器人，能够完成更为复杂的任务，适用于更多场景，比如火灾等危险场景中的救援和取物，物流业仓库物品的搬运、整理和摆放，家庭助理或老年人、残障人士助手，复杂环境和特殊对象的专业清洁，安全防爆检测，能源矿场采集，影视拍摄，甚至于送餐、送货行业等。

我们的项目主要瞄准“超市购物”这一应用场景，机器人通过超市工作人员的引导对购物环境建模，即可响应顾客在此后的各种抓取需求，极大减轻超市员工的工作负担，同时给顾客购物带来更方便快捷的体验。

## 1.2 文档概述

本文档在需求分析说明书的基础上，对系统的需求进行进一步分析设计。并对需求分析阶段出现的问题进行复盘，并建立设计与需求之间的关系，进而给出完整的项目开发设计说明书。

## 1.3 引用文档

【1】 曹泓浩.工业机器人的应用现状及发展趋势[J].科技风,2019(05):145.

【2】 晋晓飞,王浩,宗卫佳,王鹏程,王策.自主移动机器人避障技术研究现状[J].传感器与微系统,2018,37(05):5-9.

【3】 SRS103 需求规格说明书

【4】 SDP103 项目开发计划

# 2. 需求概述

### 2.1.1 业务需求描述

机器人主要使用场景为空间有限的超市内，主要任务为协助用户购物。

顾客使用之前，由超市工作人员在超市入口处启动机器人，并对其发出跟随语音指令，机器人将跟随工作人员绕超市货架行走，并建立地图数据。当到达某种指定商品货架时，工作人员发出相应记忆语音指令，机器人将该地点记录为对应商品所在地。绕行货架结束后，由工作人员带领机器人回到超市入口，发出停止跟随语音指令，机器人随即停止跟踪，在入口处等待顾客。

顾客到达超市，选择一台待命的机器人，对其发出取物的语音指令，机器人随即将当前位置记录为主人所在地，并按照顾客指令中目标物品的位置规划路径，前往指定货架识别物品和完成抓取。取物成功后，机器人规划路径返回主人所在地，把获得的物品交给顾客，完成一次取物，继续进入待命状态。

超市人员可以通过机器人机载电脑或远程控制的方式安装或卸载系统、启动或关闭机器人；通过 UI 控制机器人的启动和急停，了解机器人的运行状态；机器人将对成功获取到的语音指令及时做出相应，在到达取物地点和抓取物品成功后发出提示，并对错误和异常情况做出判断、处理和提示。

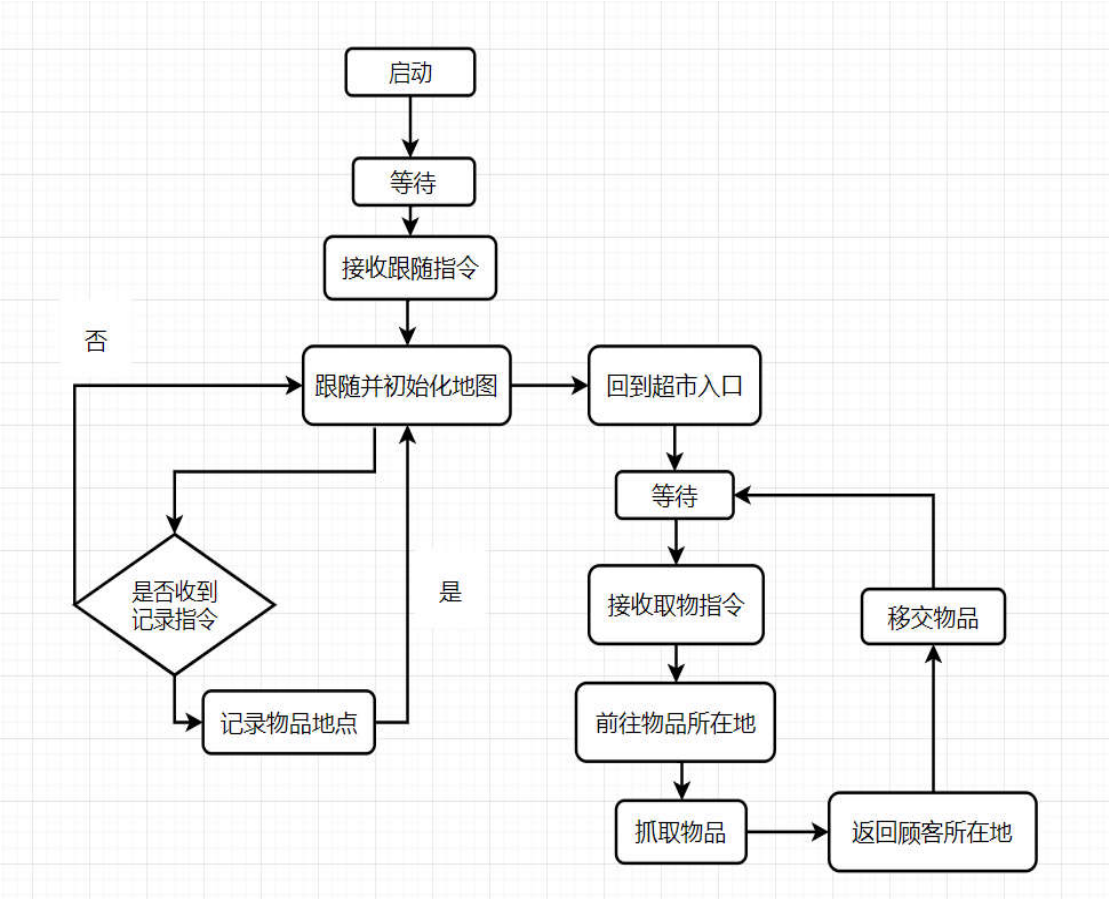



图 1 业务流图

### 2.1.2 功能需求描述

**使用者：**

- 超市工作人员
- 顾客

**使用功能描述：**

- 超市工作人员可通过机载电脑启动/关闭机器人
- 超市工作人员可通过语音指令和步行引导完成初始化地图场景建模
- 超市工作人员可以通过机载界面查看实时地图
- 超市工作人员可通过语音指令记录关键物品地点
- 顾客可通过语音命令机器人抓取指定物品
- 超市工作人员/顾客可关联错误条件

**用例图：**

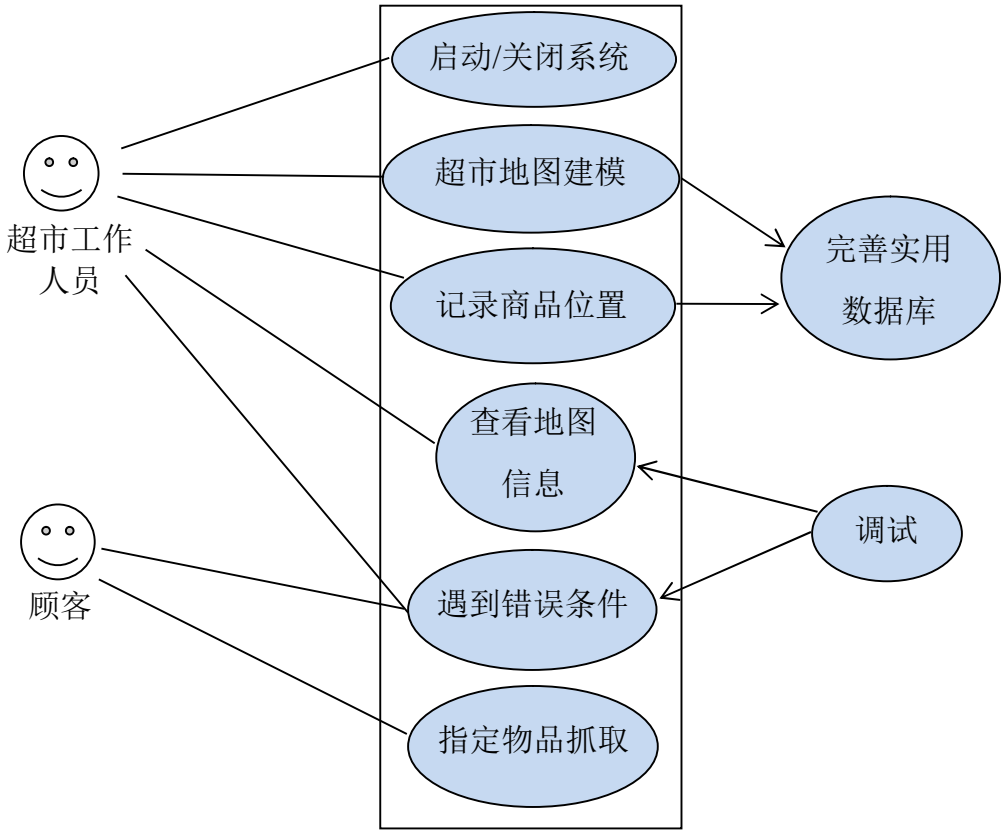



图 2 测试用例图

## 2.2 用例模型

### 2.2.1 用例一：启动机器人

**主要参与者：**超市工作人员

**目标：**使机器人系统各模块进入工作状态并确保界面可以进行交互、机器人可以接收指令

**前置条件：**机载电脑已安装本系统；超市工作人员了解规范启动步骤；超市场地平整干燥

**启动：**超市工作人员希望使用机器人

**场景：**

1. 放置机器人至超市入口处；

2. 通过 usb 接口连接机载电脑和机器人运动控件；

3. 启动机载电脑并打开机器人总开关；

4. 点击机器人系统图标，启动机器人系统；

5. 机载电脑显示系统交互界面，机器人发出语音“Startup completed”提示；

**优先级：**高

**何时可用：**第一个增量

**使用频率：**低

**次要参与者：**机器人运动控件、机载电脑

### 2.2.2 用例二：关闭机器人

**主要参与者：**超市工作人员

**目标：**关闭机器人系统，并确保下次启动时各模块状态正常

**前置条件：**机载电脑已安装本系统；超市工作人员了解规范关闭步骤；机器人处于待命状态

**启动：**超市人员暂时不再使用机器人

**场景：**

1. 关闭机器人系统界面；

2. 解除机载电脑和机器人运动控件的 usb 连接；

3. 关闭机载电脑和机器人总开关；

**优先级：**高

**何时可用：**第一个增量

**使用频率：**低

**次要参与者：**机器人运动控件、机载电脑

### 2.2.3 用例三：初始化地图场景建模

**主要参与者：**超市工作人员

**目标：**确保机器人系统正确记录超市地图

**前置条件：**机器人系统已正常启动；超市工作人员熟悉超市地形，发音标准；超市场地平整干燥

**启动：**超市人员带领机器人行进

**场景：**

1. 机器人放置于超市入口处并提示启动成功；
2. 超市工作人员走近机器人，确保底盘传感器感应到前方人员；
3. 超市工作人员以正常行走速度无折返地穿过超市货架；
4. 机载电脑显示实时地图建模情况；
5. 超市工作人员带领机器人遍历所有需要记录的货架位置；
6. 超市工作人员带领机器人回到超市入口；
7. 超市工作人员通过机载电脑界面查看场景建模情况；
8. 确保建模无误后，超市工作人员开始进行标点；
9. 机器人进入待命状态；

**异常情况：**

1. 机器人未及时跟上工作人员步伐——出现点：3，5，6
2. 场景建模不完善——出现点：7

**解决方案：**

1.工作人员发现机器人未及时跟随后，重新回到与机器人<0.5m 处，确保底盘传感器识别到前方人员，再向前走动；

2. 工作人员保证机器人硬件状态正确，完成初始化地图建模；

**优先级：**高

**何时可用：**第一个增量

**使用频率：**中

**次要参与者：**机器人运动控件、机载电脑

### 2.2.4 用例四：关键地点记忆

**主要参与者：**超市工作人员

**目标：**确保机器人系统正确记录关键商品位置

**前置条件：**机器人系统已正常启动；超市工作人员熟悉超市地形，发音标准；超市场地平整干燥；指定物品名称已录入系统数据库

**启动：**超市人员向机器人发出跟随语音指令

**场景：**

1. 机器人放置于超市入口处并提示启动成功；

2. 超市工作人员走近机器人，确保底盘传感器感应到前方人员；

3. 超市工作人员以正常行走速度穿过超市货架；

4. 机载电脑显示实时位置情况；

5. 超市工作人员带领机器人来到指定物品所在货架前；

6. 超市工作人员对准麦克风说出关键词“这是一瓶水”；

7. 机器人收到指令，记录物品位置并在地图上标注；

8. 机器人发出语音提示：“OK.I have memoried”；

9. 超市工作人员通过机载电脑界面查看物品位置记录情况；

10. 重复 19~22 步，直至所有物品位置记录完毕；

1. 超市工作人员带领机器人回到超市入口；

2. 确保建模无误后，超市工作人员对麦克风说出关键词“别跟了”；

3. 机器人收到指令，发出语音提示：“OK. What do you want me to fetch.”；

4. 机器人进入待命状态；

**异常情况：**

1. 机器人未及时收到工作人员语音指令——出现点：2，8，14

2. 机器人未及时跟上工作人员步伐——出现点：4，5，7，13

3. 物品地点记录不准确——出现点：11

**解决方案：**

1. 工作人员稍等 5~10 秒，若机器人未发出语音提示，则重新对准麦克风说出关键词，重复此步骤直到机器人响应；

2. 工作人员发现机器人未及时跟随后，重新回到与机器人<0.5m 处，确保底盘传感器识别到前方人员，再向前走动；

3. 工作人员在小范围内校准位置，使机器人跟随移动到校准地点，重新发出“memorize”指令，等待机器人响应并查看改进后的位置；

**优先级：**中

**何时可用：**第二个增量

**使用频率：**中

**次要参与者：**机器人运动控件、机载电脑

### 2.2.5 用例五：指定物品抓取

**主要参与者：**顾客

**目标：**命令机器人到对应货架取回指定物品

**前置条件：**超市地图初始化已完成，指定物品位置已正确记录；机器人处于待命状态；顾客了解关键商品名称，发音标准；超市场地平整干燥

**启动：**顾客向机器人发出取物指令

**场景：**

1. 机器人放置于超市入口处并处于待命状态；
2. 顾客对准机器人麦克风说出目标物品关键词，如“我想要一杯**可乐**”；
3. 机器人收到指令，发出语音提示：“Ok.I will go to get it for you.”；
4. 机器人将当前位置记录为顾客所在地，并在地图上显示；
5. 机器人规划前往指定物品所在货架的路径；
6. 机器人开始移动；
7. 机器人到达指定物品所在货架；
8. 机器人摄像头识别物品并校准位置；
9. 机械臂启动，并开始抓取目标物品；
10. 目标物品抓取成功，机器人发出语音提示：“I got it.I'm coming back.”；
11. 机器人规划前往顾客所在地的路径；
12. 机器人开始返回顾客所在地；
13. 机器人到达顾客所在地；
14. 机械臂松开，将物品移交给顾客；
15. 机器人发出语音提示：“Ok.What do you want next?”并进入待命状态；

**异常情况：**

1. 机器人未及时收到顾客语音指令——出现点：2
2. 机器人未识别到用户指定物品——出现点：2
3. 指定物品位置还未在地图上登录——出现点：5

4. 物品抓取不成功——出现点：10

5. 物品抓取类型错误——出现点：15

6. 机器人在行进过程中遇到突发障碍物——出现点：6，13

**解决方案：**

1. 顾客稍等 5~10 秒，若机器人未发出语音提示，则重新对准麦克风说出关键词，重复此步骤直到机器人响应；

2&3. 机器人无反应，顾客重新根据已有关键词发出指令，直到机器人响应；

4&5. 机器人无反应，并返回顾客所在地，顾客重新发出指令；

6. 机器人在到达障碍物之前减速至停止；

**优先级：**中

**何时可用：**第三个增量

**使用频率：**高

**次要参与者：**机器人运动控件、机载电脑、目标物品

## 2.3 非功能性需求

### 2.3.1 系统可靠性需求

系统应保证超市营业时间内不间断运行，系统硬件构成应具有冗余等安全措施。设备的 MTBF（Mean Time Between Failure，平均故障间隔时间）应小于或等于每年 50 分钟内。每年每台设备故障率不超过 2%。设备具有避免单点失效的功能，从而保证系统的可靠度不低于 99.99%，要求某一个设备的宕机不会影响业务的运行。

系统应具备软件、硬件故障在线恢复的能力。重大故障时间间隔应大于 1 年，故障平均修复时间小于 1 小时。

需配置冗余热备份的电源模块，所有电源模块支持-48V 直流供电和 220 交流电，可按需配置。系统须配置冗余热备份的风扇散热系统。

设备应采用具有电信级高可用性的操作系统。

机器人设备支持命令行和控制台界面，易于配置和管理。

### 2.3.2 系统可扩展性需求

机器人设备应具备良好的可扩展性，能够适应系统容量的扩大和管理内容的增加，包括软硬件平台、系统结构、功能设计、管理对象。 随着管理功能的增加，要求系统具有灵活的扩展性。

### 2.3.3 系统易用性需求

易用性是一种以使用者为中心的设计概念，易用性设计的重点在于让产品的设计能够符合使用者的习惯与需求。

在本机器人系统中，我们希望让使用者在使用该机器人的过程中不会产生压力或感到挫折，并能让使用者在使用机器人的各项功能时，能用最少的努力发挥机器人最大的效能。

### 2.3.4 系统安全性需求

系统保密性需求：只有授权的用户才能动用和修改信息系统的信息，而且必须防止信息的非法、非授权的泄漏。

系统完整性需求：也就是说信息必须以其原形被授权的用户所用，也只有授权的用户才能修改信息。

漏洞检测和安全风险评估：识别检测对象的系统资源，分析这一资源被攻击的可能指数，了解支撑系统本身的脆弱性，评估所有存在的安全风险

系统可用性和抗毁性：设备备份机制、容错机制，防止在系统出现单点失败时，系统的备份机制保证系统的正常运行。

系统防病毒：防病毒系统应基于策略集中管理的方式，使得分布式的企业级病毒防护不再困难，而且提供病毒定义的实时自动更新功能。

# 3. 体系结构设计

## 3.1 总体结构

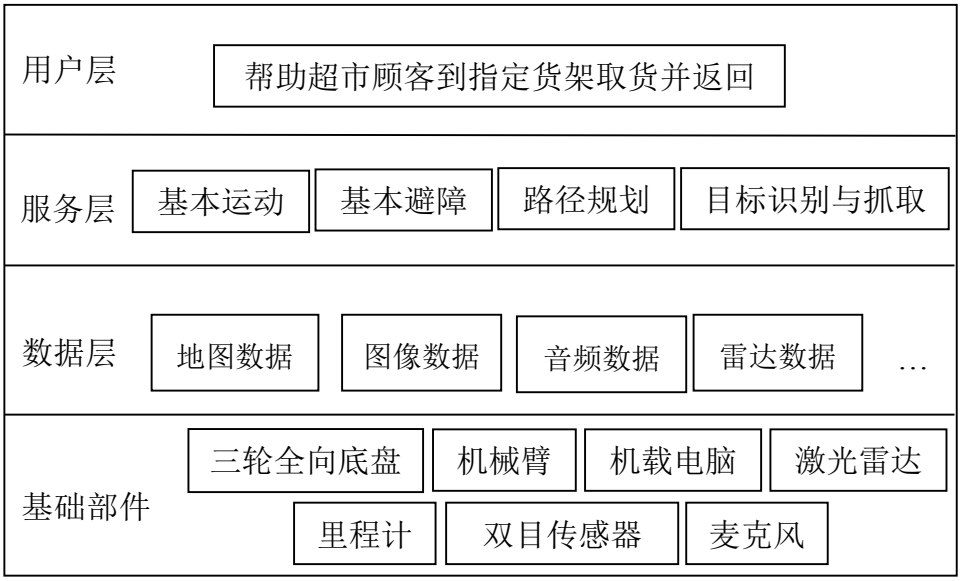



图 3 总体结构图

### 3.1.1 软件体系结构

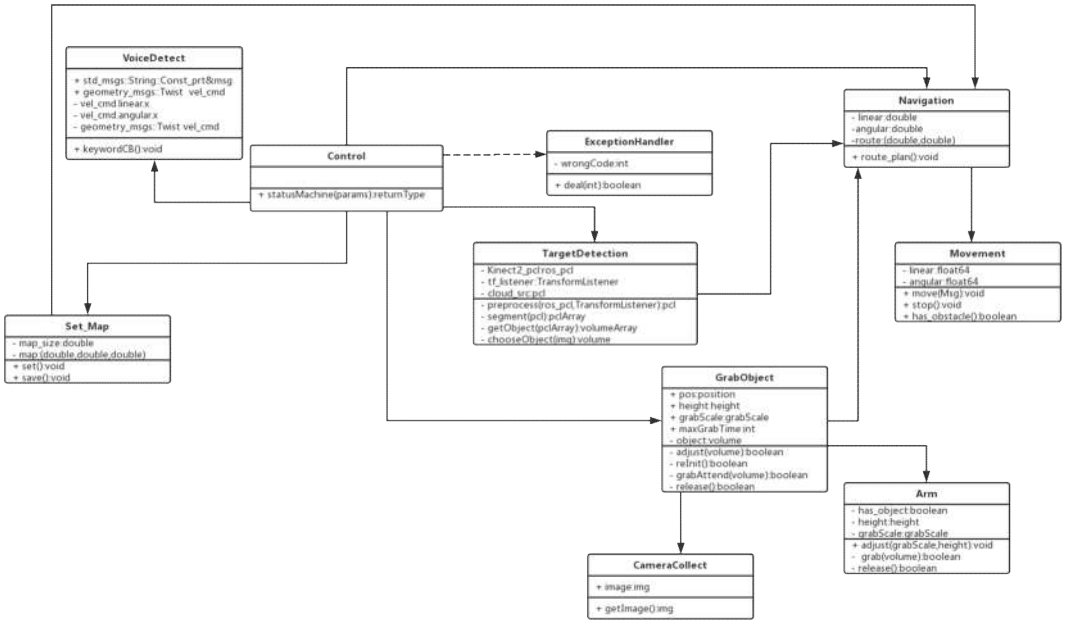



图 4 软件体系结构

系统类图如上所示。Control 类为控制类，也是整个系统的主类。语音识别类 VoiceDetect 在系统运行过程中不断监听，并实时返回捕获到的指令给 Control。Control 类含有一个状态机，语音识别类返回的指令需要根据状态机当前状态来判断是执行还是丢弃。系统初始时管理员建图需要发出相应的语音指令，触发主控调用 Set_Map 类相关方法完成建图与存储。建图过程中还会调用 Navigation 类启动路径规划，具体的运动实现由 Movement 类承包。之后用户可以语音指挥机器人去取物，取物需要用 TargetDetection 识别特定物体并返回该物体的相关参数。识别特定物体需要 TargetDetection 类调用 CameraCollect 双目摄像头采集的图像。之后主控再调用 GrabObject 类不断尝试抓取物体，需要调用 Arm 机械臂完成抓取参数的调节。尝试完毕之后调用 Navigation 类回到顾客所在的地方，如果抓取成功调用 GrabObject 的释放方法释放物体。物体检测、抓取与释放之前都会调用 Navigation 以使得能够对准物体。系统运行过程中如果出现异常，各个类将会通知 Control，Control 调用 ExceptionHandler 的处理方法处理各类问题。

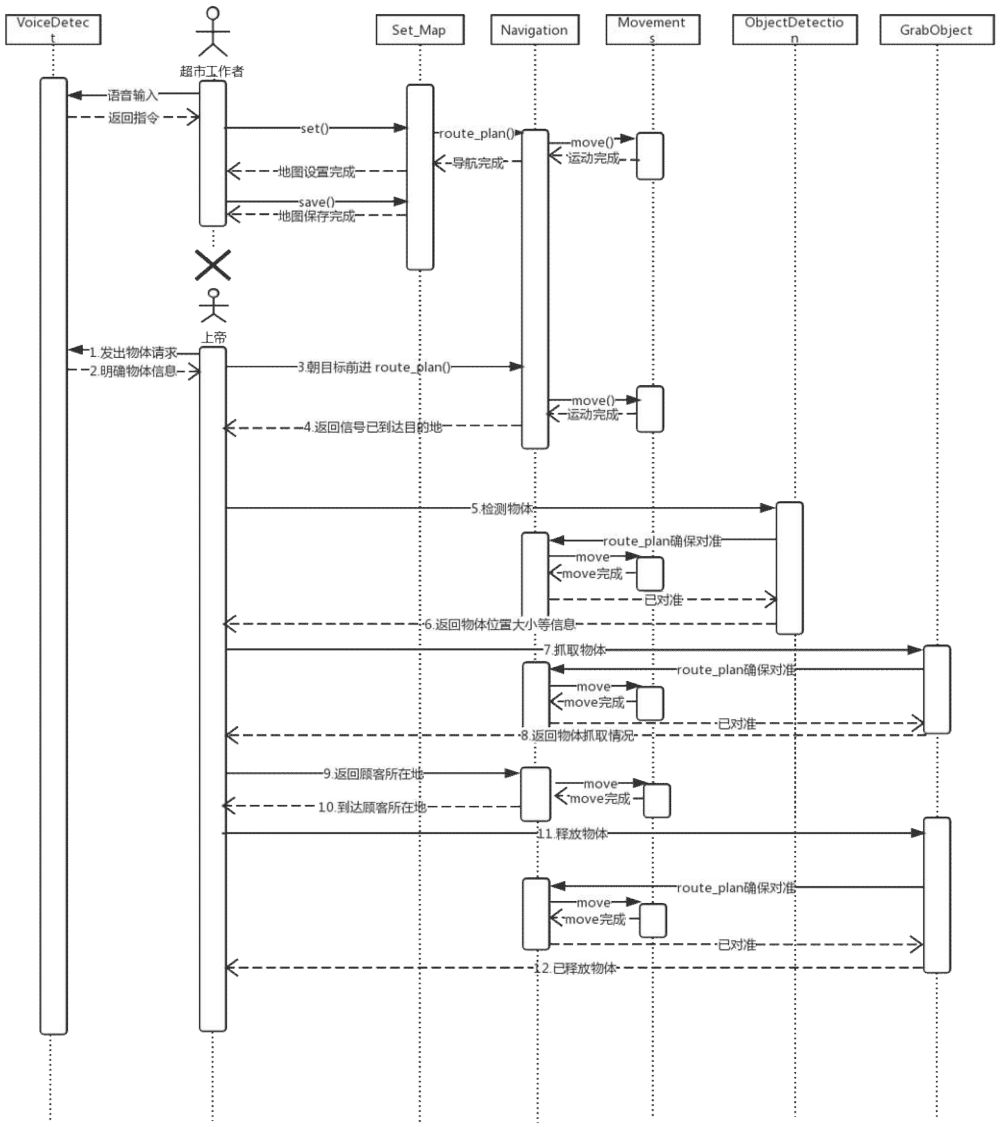



图 5 时序图

### 3.1.2 硬件体系结构

用户通过语音输入命令，机器人会调用语音模块解析语音，返回给主控模块，进行功能选择，然后可以执行建图功能或者取物功能。两种功能会分别调用系统的主要几个模块，同时各个模块之间也会有相互通信。每个模块最终会通过硬件接口实现具体功能。

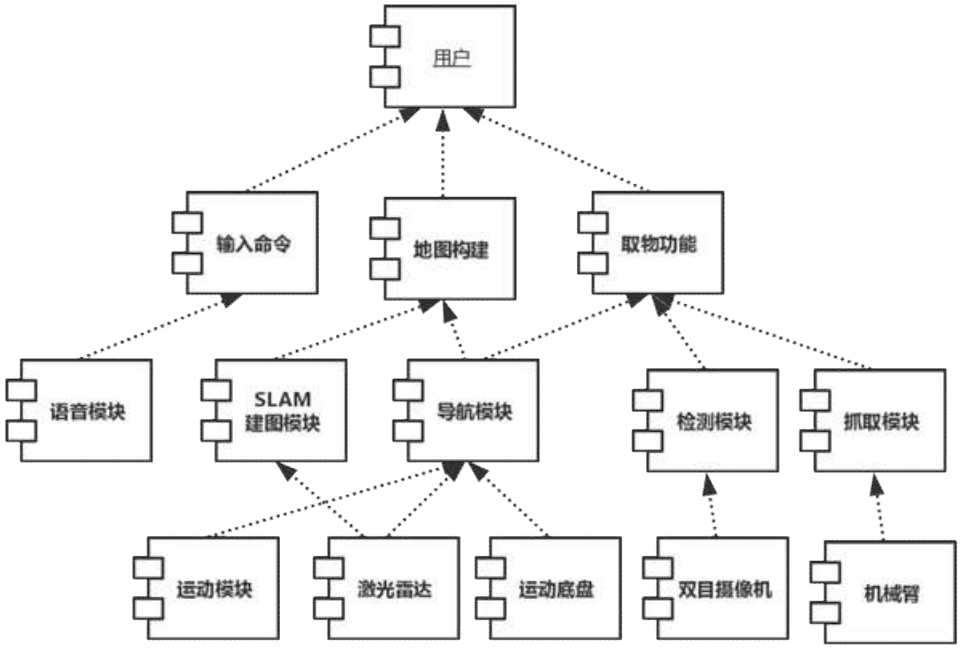



图 6 系统构件图

### 3.1.3 支撑体系

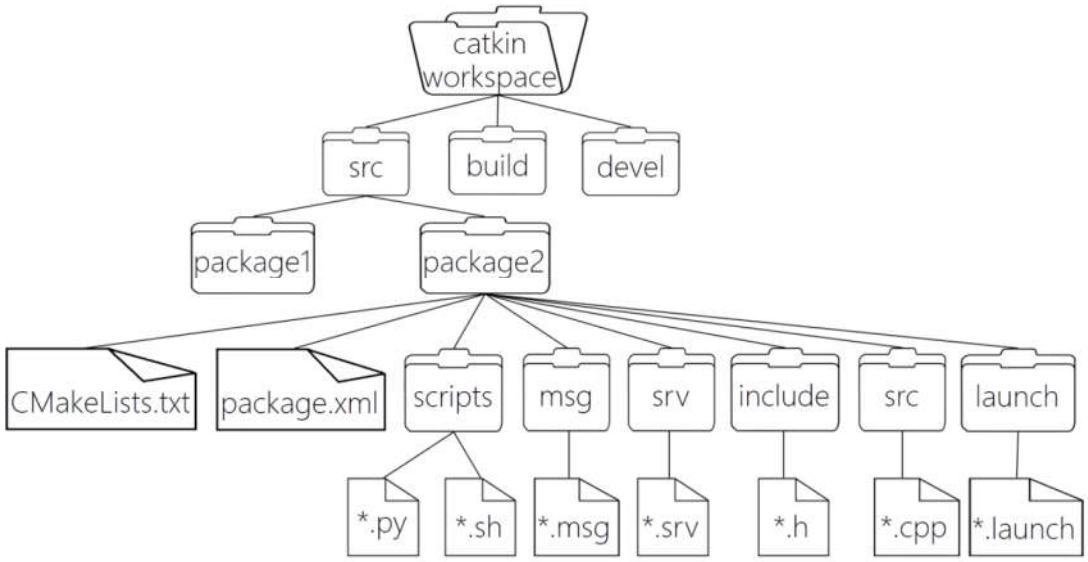



图 7 支撑体系

## 3.2 关键问题及解决方案

### 3.2.1 软件总体结构的组织

为了便于后期团队成员在对系统的维护和升级，因此如何设计一个便于团队成员沟通的软件总体结构是十分必要的。

为此我们的软件总体结构，从大体上看，呈现金字塔形。即从顶层功能出发，划分出若干子功能模块，将逐个模块的功能进一步细分，呈现一个自顶向下的设计架构。

采用金字塔模型的好处在于，各功能模块的联系紧密，逻辑关系也更加清晰，因此降低了团队成员沟通协作上的难度，提高了开发效率。

### 3.2.2 模块的共享与复用

模块的共享或复用可以大幅提高编程的效率，省去了大量“重复造轮子”的人力成本和时间成本，但是实际上很多时候我们想复用却复用不了，或者复用本身带来了比较大的成本。

对于机器人系统架构而言，复用的关键因素在于对于问题和解决方案适当抽象，具体来说，一方面我们需要确保机器人系统所需要的功能都能通过抽象接口展示出来，也就是说接口的功能必须是完整的，同时这个抽象能够屏蔽掉无关的细节，也就是尽可能地简化。

由于机器人功能需求的具体实现错综复杂，并且我们在描述需求的时候有很多模糊的名词，所以一个常见的误区在于我们通常会混淆两个看起来类似，但是有不同实质的问题。在系统的实际实现过程中，我们会尽量保证名字相近的不同的问题不会放到一起去考虑，尽量减小功能复用的二义性。

### 3.2.3 系统总体结构设计的核心与边界

在系统总体结构设计的过程中，一个常见的问题就是在设计上求大求全，却没有清晰的关于问题核心和边界的概念，常常突破产品所在的抽象层次，看起来是带来了更多的功能，实质上却是破坏了整个架构的设计，模糊了各个概念的边

界。由于潜在使用者无法理解这样的组件对应的问题，同时也担心其潜在的复杂逻辑关系，导致了这样的组件反而很难被复用。

在我们的机器人系统架构设计中，问题的核心在于建立起机器人各组件之间功能上的联系，并将模块集成后的系统作为一个整体发布给用户。问题的边界在于对机器人性能上的优化控制，包括但不限于机器人异常状态处理、提高检测识别精度等等。只有在服务化的架构理念里，当机器人系统设计能够给出一个尽可能简单而有效的服务目标时，才更有可能在更广阔的场景下使用这个服务。

# 4. 接口设计

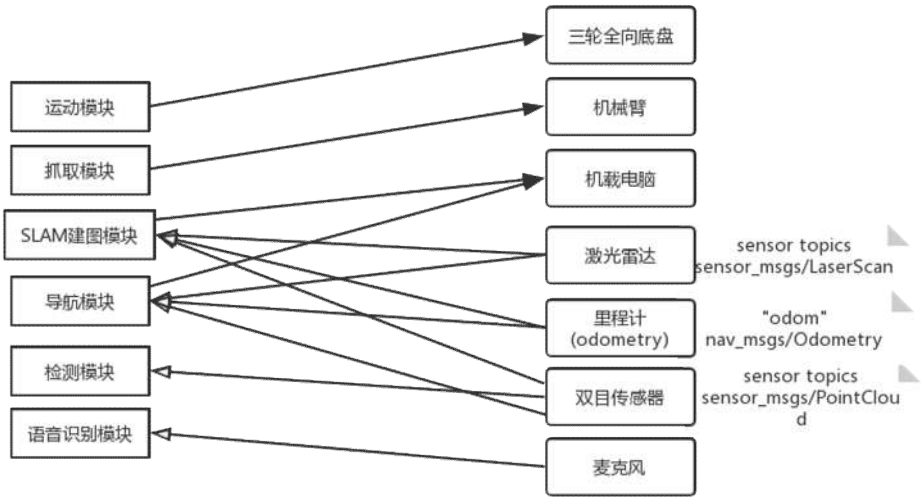



**图 8 系统软硬件结构设计**

运动模块为三轮全向底盘提供速度角速度信息，抓取模块为机械臂提供抓取参数信息，SLAM 建图模块与导航模块共同为机载电脑提供地图信息。

激光雷达为 SLAM 建图模块与导航模块提供 sensor_msgs/LaserScan 信息，里程计(odometry)为 SLAM 建图模块与导航模块提供 nav_msgs/Odometry，双目传感器为 SLAM 建图模块、导航模块与目标检测抓取模块提供 sensor_msgs/PointCloud 信息，麦克风为语音识别模块提供检测到的语音指令信息。

# 5. 数据库设计

数据库是软件开发的重要组成部分，由于数据流的规范约束了程序不同模块之间的相互协同配合，因此软件开发过程中的数据库设计显得尤为重要。

在本项目所使用到的数据库，是由若干个数据实体以及实体之间的关系构成的，通过对数据实体的数据项以及数据实体之间的关系的分析，来对本项目的数据库进行设计。

## 5.1 实体概述

### 5.1.1 机器人运动参数

交互界面是用户用来启动、控制机器人系统的 UI，用户可以通过对设置项中数据的更改，完成对机器人软件的配置。

### 5.1.2 地图

地图数据通过机器人系统初启动过程完成建立与存储。地图数据为后期的路径规划、导航功能提供必要的支持。

### 5.1.3 关键词库

关键词库存储了语音功能所能识别的用户语音指令。关键词库为语音识别功能提供必要的支持。

### 5.1.4 抓取目标

存储指定的抓取目标的信息，如宽、高、位置信息，提供信息帮助机器人靠近目标、规划路径、接住目标 。

### 5.1.5 机械臂运动参数

机器人的机械臂受到机器人核心系统的控制，实现对物体的抓取功能。该项数据需要记录机械臂的上升高度，闭合宽度等，用于实现对物体的抓取。

### 5.1.6 异常处理情况

对于各种异常情况，如机器人出现侧翻、用户的目标指定使机器人遭遇危险等进行处理。

## 5.2 ER 图

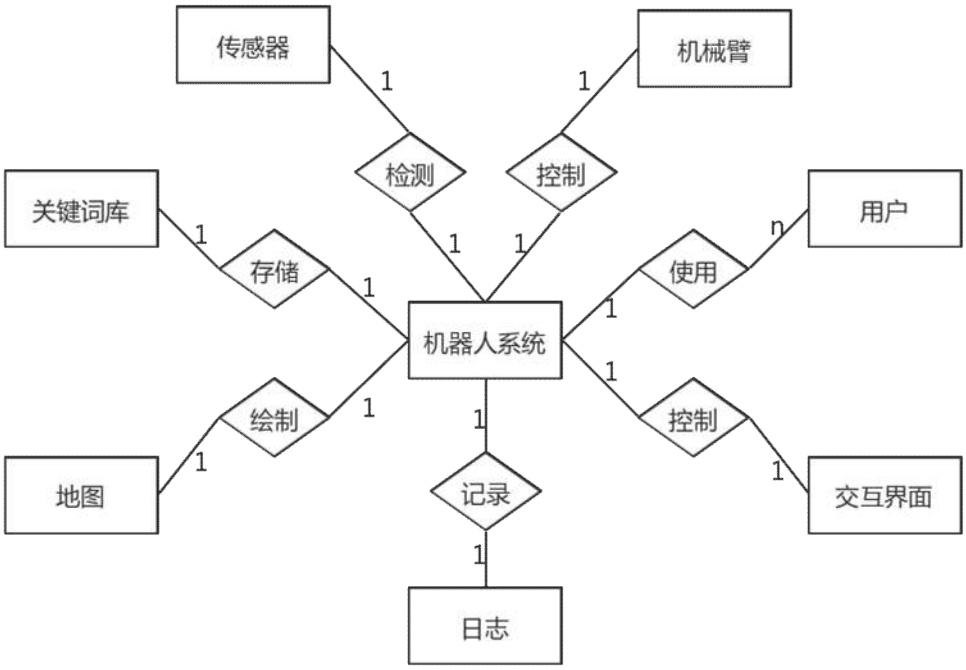



图 9 ER 图

# 6. 详细设计

ROS 对机器人所有模块的控制都是向特定目标广播特定类型的数据来实现的。具体实现主要分为三个步骤：

1. 通过 ros::NodeHandle 类型的对象，来生成一个广播对象，比如 pub，同时需要指定广播的目的位置。
2. 创建对应模块的特定类型对象，与广播对象设置的目的位置对应一致。并设置好该对象的数据。比如运动模块使用 geometry_msgs::Twist 类型数据，向 /cmd_vel 广播。

3. 使用 ros::Publisher()方法发布数据

通过上面三个步骤就可以控制机器人某个模块，所以我们的任务就是将各个模块有机结合，实现一套复杂的功能。我们的系统有六大模块：运动模块，SLAM 见图模块，导航模块，检测模块，抓取模块，语音识别模块。

## 6.1 运动模块

【功能】

运动模块主要负责控制机器人实现基本的运动，主要是平移运动和原地旋转运动。运动模块可以接收位移向量作为输入参数，然后规划机器人的速度和运动时间开始运动。另外运动模块需要接收激光雷达的测距信息，从而对障碍物做出急停反应。

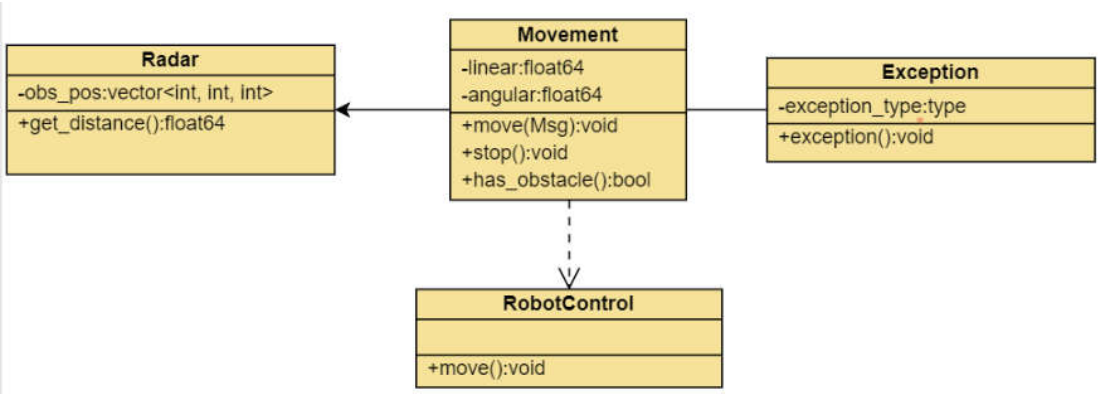
【输入】

Msg 类型数据，内含机器人的 xyz 三个轴的速度和角速度。

【输出】

运动控制指令，设置机器人的速度和角速度直接调用内置模块控制移动。

【设计】




模块的主类是 Movement，它调用机器人控制系统的方法来操作机器人的运动，这里内置接口用 RobotControl 类来表示。运动类的主要工作就是设置机器人的速度和角速度，具体运动过程的规划由上层模块来实现。

运动模块还要通过 Radar 类来检测运动路径上是否有障碍物，可以通过障碍物距离信息来判断，从而保证安全移动。另外还需要异常处理类 Exception，对运动过程中的异常进行处理。

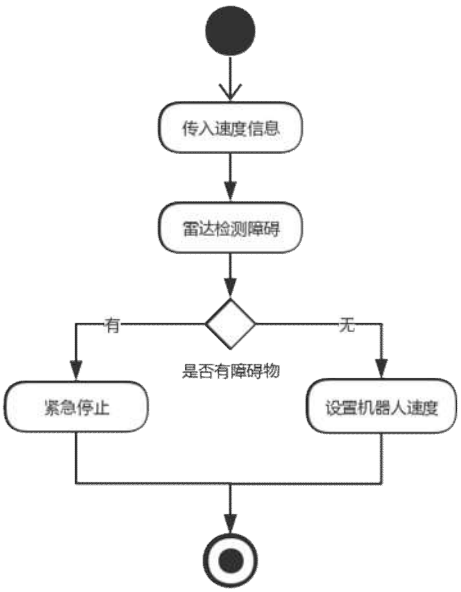



图 11 运动模块流程图

## 6.2 SLAM 建图模块

【功能】

机器人每到一个新场景后，实现机器人在跟随超市管理员绕场一周的过程中即时反馈自身定位与雷达检测到的障碍信息，完成并保存对超市地图场景的建模。

【输入】

机器人调用 amcl 获取的自定位置信息和使用 tf 发布的有关坐标框架之间关系的信息；

传感器通过 ROS 发布的 sensor_msgs/LaserScan 或 sensor_msgs/PointCloud 信息；

使用 tf 和 nav_msgs/Odometry 发布的测距信息；

【输出】

建好的超市地图场景(map.pgm 和 map.yaml)

【设计】

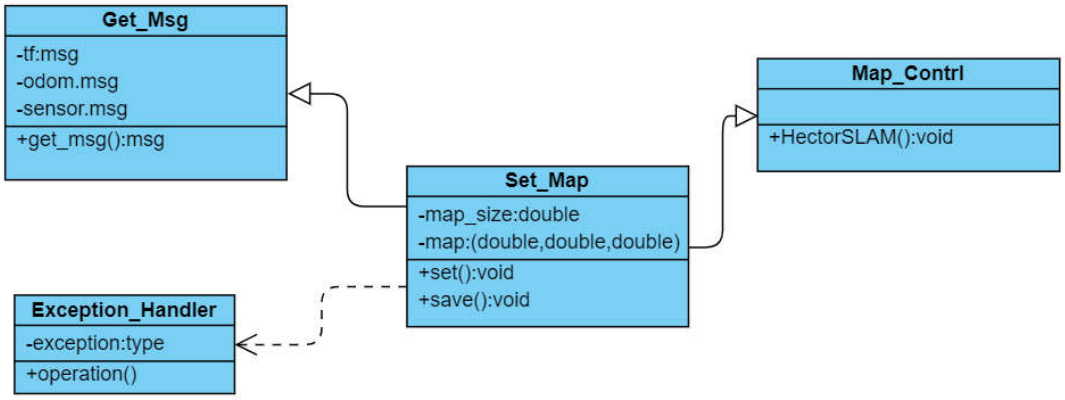



图 12 SLAM 建图模块类图

Get_Msg 类为软硬件交互类，通过 get_msg 方法实时获取 amcl 的自定位信息，测距信息及传感器信息等硬件反馈的信息。Set_Map 为总控类，它接受 msg 信息，使用 set 方法完成对地图的建模，并通过 save 方法保存地图。Map_Contrl 类控制整个建图过程，主要是实现 HectorSLAM 算法建图。设置一个异常处理类，当建图过程中发生异常，调用异常处理类进行异常处理。

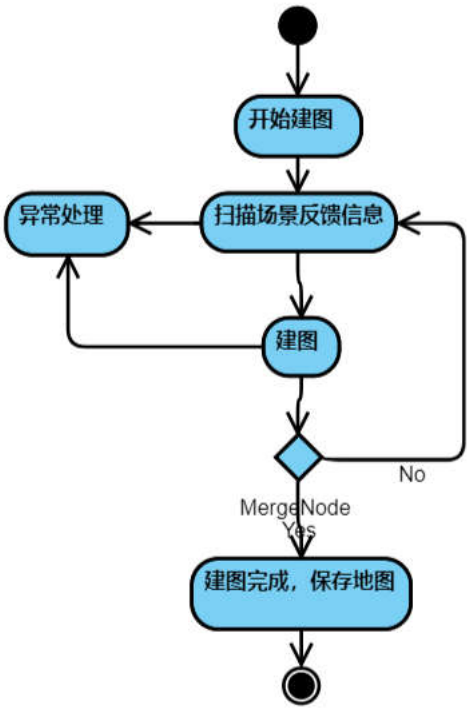



图 13 SLAM 建图模块流程图

## 6.3 导航模块

【功能】

通过获取需要的数据流实现机器人的路径和行为规划。并将导航需要用到的各个部分联系在一起。

【输入】

机器人调用 amcl 获取的自定位置信息和使用 tf 发布的有关坐标框架之间关系的信息；

传感器通过 ROS 发布的 sensor_msgs/LaserScan 或 sensor_msgs/PointCloud 信息；

使用 tf 和 nav_msgs/Odometry 发布的测距信息；

完成 SLAM 建图后获取的 map 信息；

机器人当前需要完成的目标信息；

【输出】

机器人的 x 速度，y 速度， θ 速度；

整体路径规划得到的路径；

【设计】

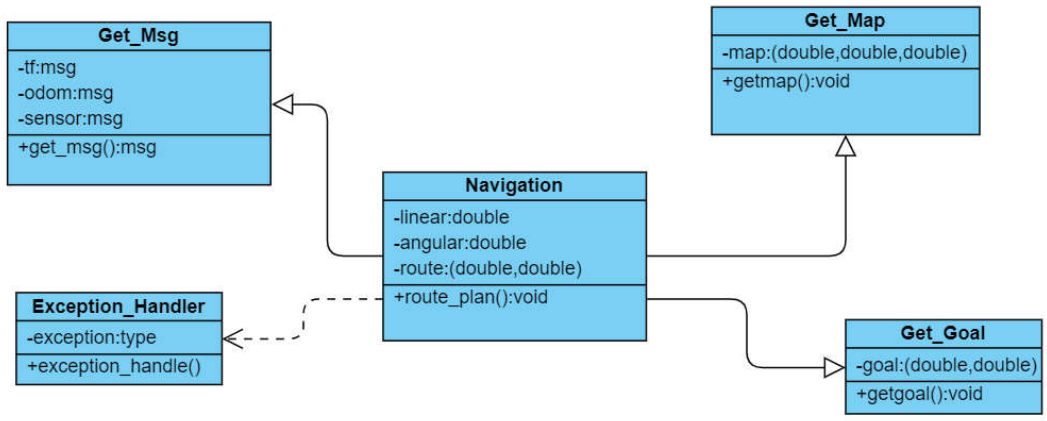



图 14 导航模块类图

Get_Msg 类为软硬件交互类，通过 get_msg 方法实时获取 amcl 的自定位信息，测距信息及传感器信息等硬件反馈的信息。Get_Map 类通过 get_map 方法从 SLAM 建图模块获取地图信息。Get_Goal 类通过 getgoal 方法从总控模块获取处理好的目标地点信息。Navigation 类为导航模块的总控类，它接受 map 信息，goal 信息，并实时接收 msg 信息，通过 route_plan 方法调用 ROS 中的 Navigation 导航堆栈实现总体与实时路径规划，得到机器人整体规划路径，并得到每个时刻的

速度和角速度。设置一个异常处理类，当路径规划过程中发生异常，调用异常处理类进行异常处理。

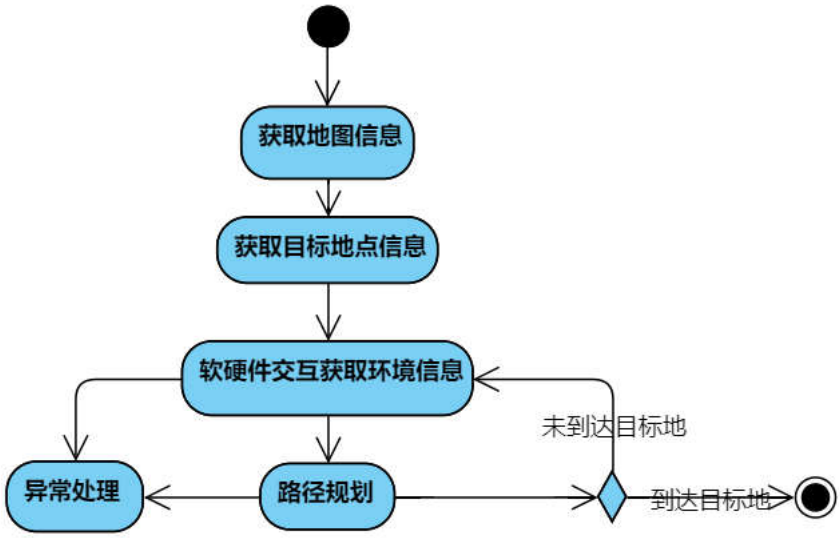



图 15 导航模块流程图

route_plan 方法调用 ROS 中的 Navigation 导航堆栈实现如下：

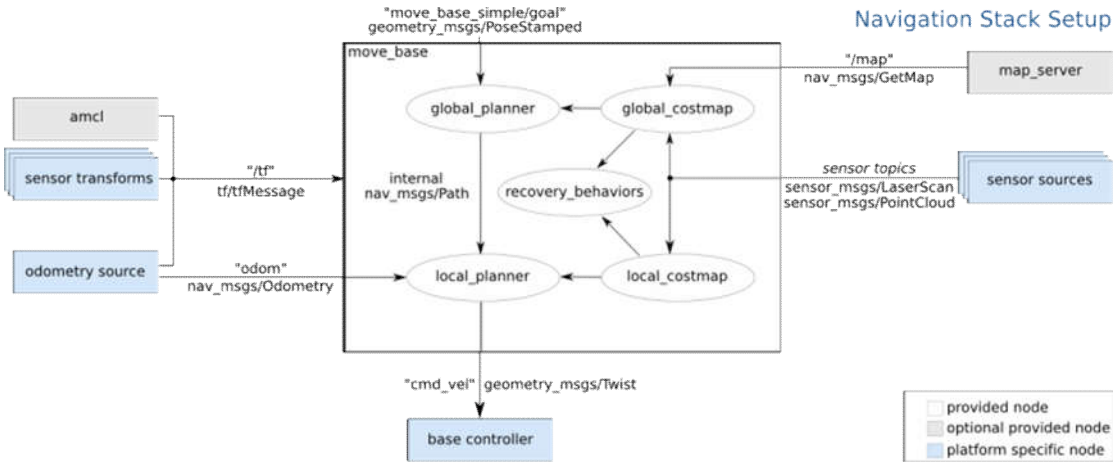



图 16 导航模块堆栈实现图

在路径规划的实现方法中，使用两个成本映射来存储获取到的有关地图上障碍物的信息。一个成本图用于全局规划，即在整个环境中创建长期计划，另一个用于本地规划和避障。

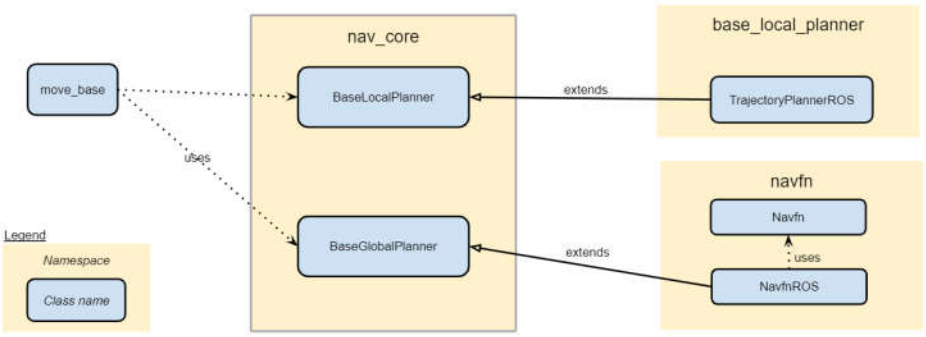



图 17 导航模块成本图

由于 ROS 中的 Navigation 导航堆栈是已实现的，上述 Navigation 具体内容我们只需理解并将它封装调用即可。

## 6.4 检测模块

【功能】

检测模块可以对物品位置进行检测，通过利用 Kinect2 视觉传感器，实现相应功能。需要预先对 Kinect2 视觉传感器进行适配和校准。

【设计】

物品检测功能的实现流程如下：

1.检测预准备

检测的预准备过程中需要创建用于坐标转换的 tf_listener，订阅 Kinect2 的点云数据，创建和发布用于展示检测结果的主题。

检测预准备工作完成后，模块进入检测等待状态，随时可以对上传的点云数据，并对相应数据进行分析处理。

2.预处理

预处理包括点云坐标转换和数据格式转换。点云坐标转换的任务是将点云坐标值从基于 Kinect2 传感器转换成基于机器人地面投影中心，数据格式转换的任务是将 ROS 格式数据转换成 PCL 格式数据。

经过预先处理之后，就能直接使用 PCL 的平面检测算法进行分析检测。

3.点云分析检测

点云分析检测使用 PCL 的平面检测算法进行点云的分析检测。

PCL 的平面检测算法首先使用分割对象 segmentation 将水平平面检测出来。

再遍历所有平面，找出高度符合要求的平面作为标准平面。随后去除标准平面内的点云，将标准平面上方一定距离之内的点云分离出来，作为物品点云集合；最后用 Kd-Tree 对物品点云集合进行近邻搜索查找，将互相分离的点云团簇分割出来。每个团簇认为是一个物品。

4.生成检测结果

检测结果实现了将识别到的点云簇转换为具体的物品，从而得到物品的形状、体积，并对物品进行标号。

5.发布检测结果

这里的工作是完成检测结果的发布，主要通过步骤 1 中创建的检测结果 topic 来完成。检测结果发布之后，其他模块就能通过相应 topic 获取需要的结果信息。

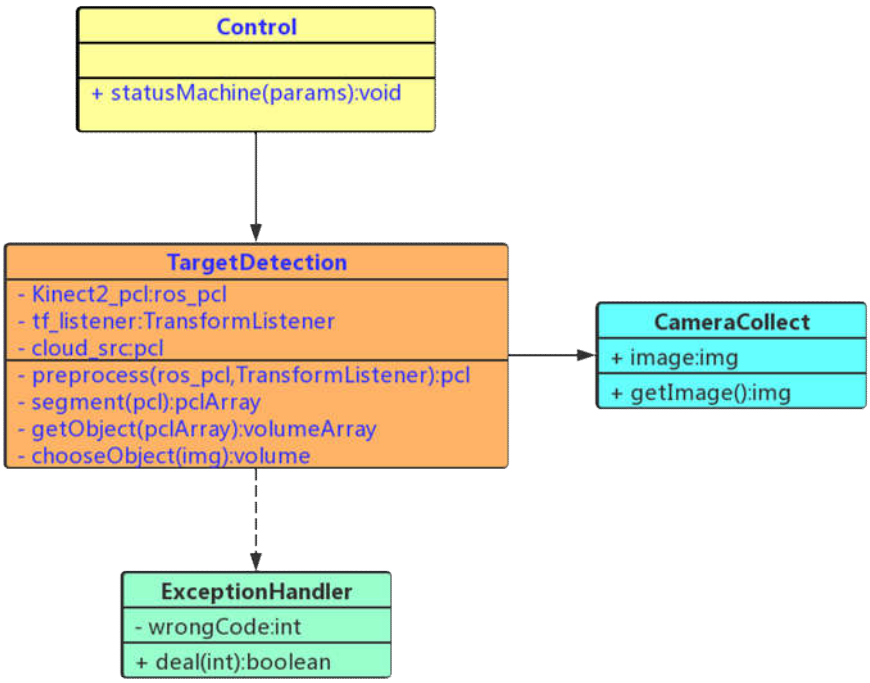



图 18 检测模块类图

【类图说明】

检测模块的主类是 TargetDetection，控制类通过 TargetDetection 实现物体检测与识别。主类的 preprocess 方法将初始出 TDPreprocess 类，通过 kinect2base 方法进行坐标转换，通过 ros2pcl 方法进行数据格式转换。主类 segment 方法将利用 TDSegment 中 planeDetect 方法得到平面点云，之后利用 getObjectCld 筛去

平面点云得到其上方物体点云集合。主类的 getObject 方法调用 TDCld2Object 类中 neighborSearch 方法对物品点云集合进行近邻搜索查找，将互相分离的点云团簇分割出来。之后调用同类的 volumeStatic 方法完成从点云团簇到物品的转换，得到物品的形状、体积。最后主类通过调用 chooseObject 方法结合 CameraCollect

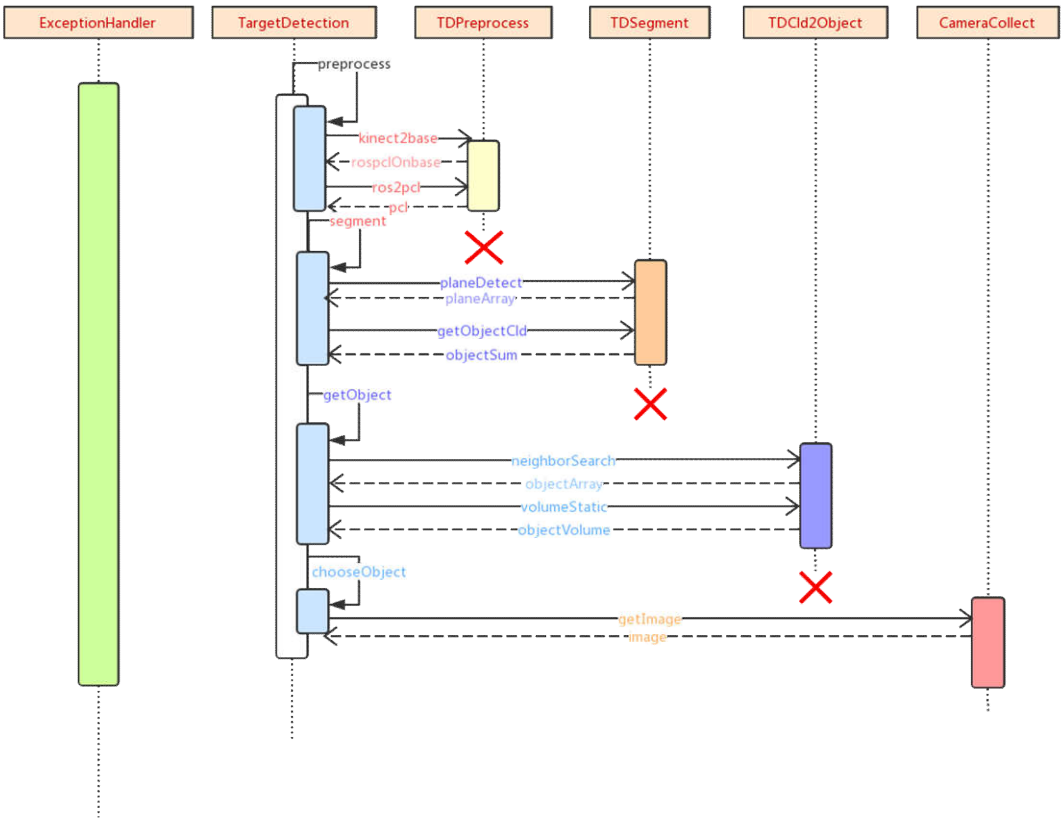



图 19 检测模块流程图

中返回的信息完成识别特定物体的工作。ExceptionHandler 对所有出错情况进行异常处理。

## 6.5 抓取模块

【功能】

抓取模块利用了物体检测得到的结果，包括平面检测结果、物体检测结果。之后进行抓取操作、放回操作。在调用抓取模块之前需要进行一系列的适配操作，最主要的是调节机械臂参数。

【输入】

1. 控制到平面的距离

得到检测模块的平面识别结果之后，机器人通过调用运动模块，前后运动控制到平面的距离，方便下一次物体检测和抓取。

2. 控制至待检测物体的距离

调用检测模块得到物体检测的结果，挑选出待抓取的目标物体。调用运动模块使得机器人和待抓取物体之间的距离适中。

3. 进行抓取

机器人伸出手臂，控制手爪闭合宽度以抓取物体，拿起物体，完成抓取任务

4. 放置物体

机器人带着物体回到用户处，张开手爪、放开物品。

5. 机械臂状态复原

完成抓取任务之后，机器人会调整机械臂到抓取之前的状态（也是默认的抓取待命状态）。这保证了机器人每次抓取之前的机械臂状态的一致性。

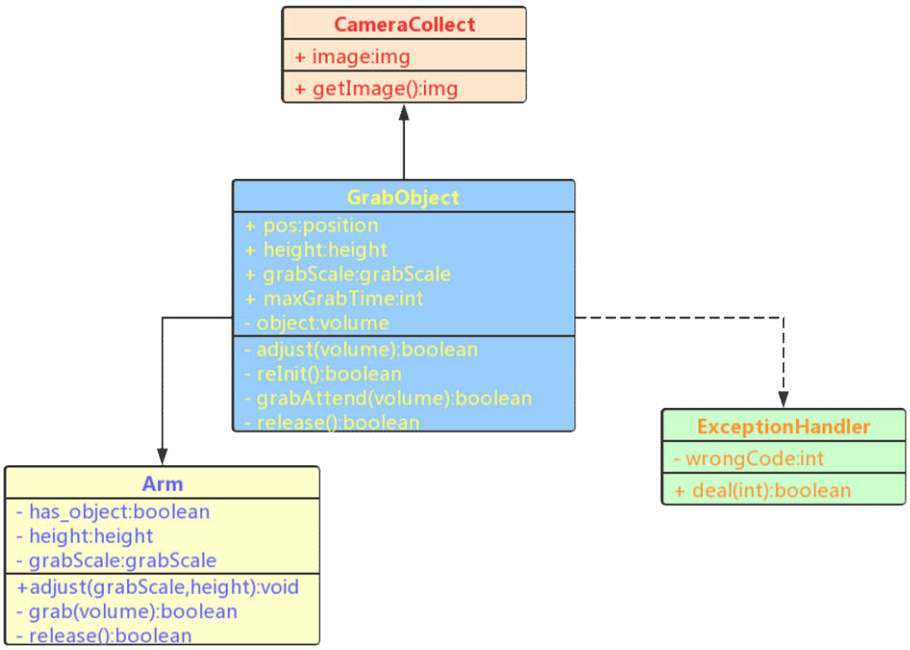



图 20 抓取模块类图

【类图说明】

抓取模块的主类为 GrabObject 类。adjust 方法根据物体的形状大小位置，调用 Arm 类 adjust 方法调节机械臂的高度、抓取参数。grabAttend 方法调用 Arm

类 grab 方法。release 方法调用 Arm 的 release 放开抓取物体。抓取的最后可以调用 reInit 方法回到抓取之前的状态，保持前后一致性。ExceptionHandler 处理异常情况。

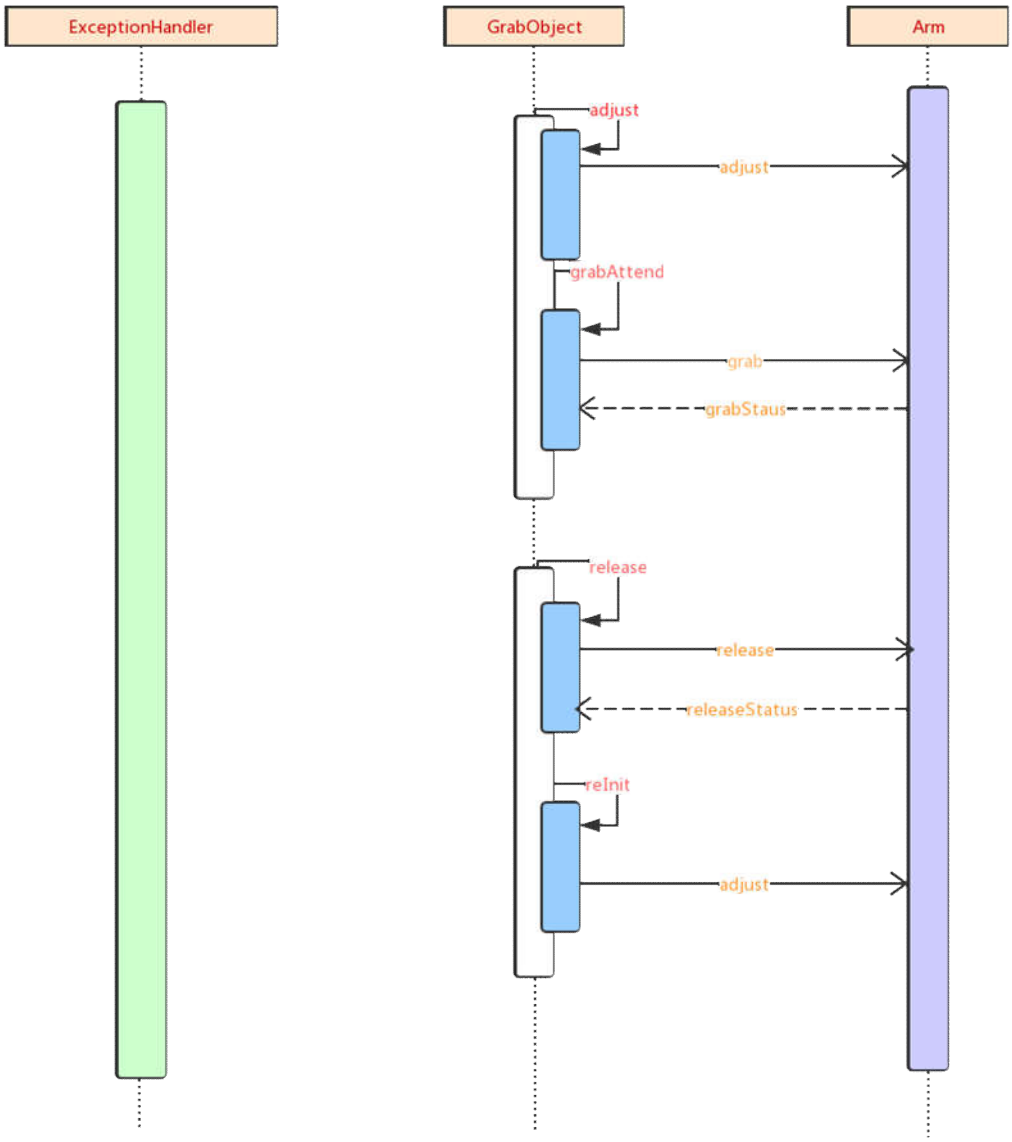



图 21 抓取模块流程图

## 6.6 语音识别模块

【功能】

语音识别关键词，可以对用户的语音指令进行识别，并完成关键词库的扩展与数据库录入存储。扩展用户输入指令集，由于识别结果为字符串形式发布，因此可以建立指令与识别到的字符串之间的映射关系，从而达到指令预期的执行效

果。机器人可以进行语音播报，与用户进行交互，向用户确认指令信息和系统状态。

【输入】

模块输入端通过用户的语音指令直接完成数据获取。

【输出】

模块返回识别到语音指令的字符串。

【设计】

语音识别：启动 PocketSphinx 脚本，将识别出的语音内容转化成字符串形式发布。修改.corpus 文件，训练得到新的字典文件和模型文件，替换掉源文件并覆盖，只支持英文关键词输入。重新 launch，检查结果即可。或调用科大讯飞提供的库，完成上述过程。

语音播报：引入 ROS 系统头文件和 sound_play::SoundRequest 头文件，定义一个 sound_play::SoundRequest 对象 sp 设置对象的参数，使用 tts_pub 进行发布朗读，使用 sleep()控制循环周期，使用 while(n.ok())进行循环控制。通过触发条件判断，当条件触发时，将 sp.arg 的值修改为需要播报的语句，重新 cmake 并 launch 即可。

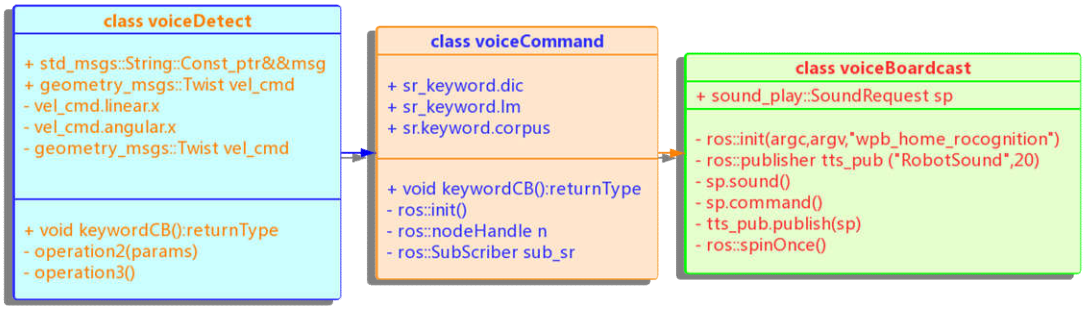



图 22 语音模块类图

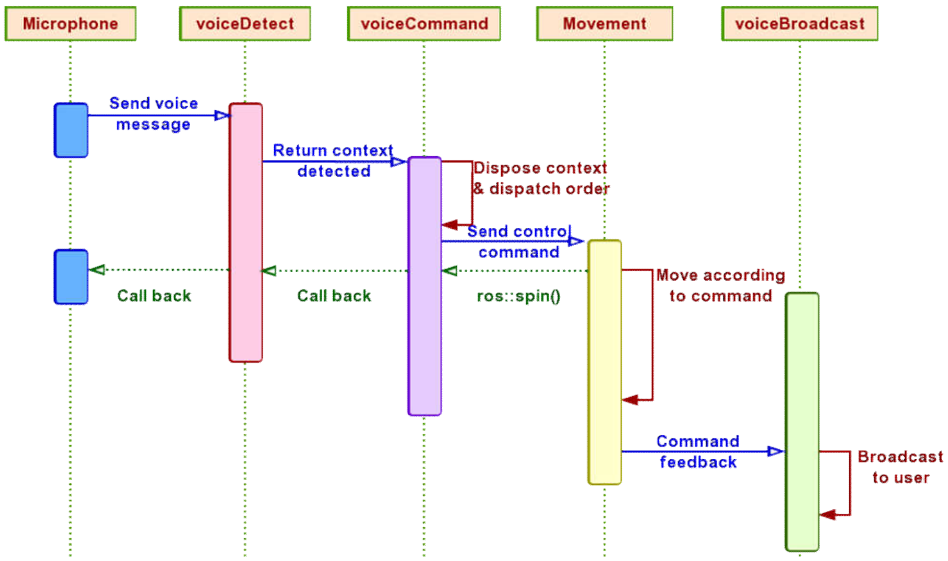



图 23 语音模块流程图

# 7. 运行与开发环境

## 7.1 运行环境

### 7.1.1 硬件环境

嵌入式开发板+各类传感器+运动装置+机械臂装置

### 7.1.2 负载能力

启智 ROS 机器人重量约为 30kg（包含抓取组件），承载能力 10kg。

### 7.1.3 工作环境

启智 ROS 机器人是室内机器人，在此环境之外运行可能会损坏机器人。工作平面需要能够承载不小于 40kg 的重量。如果表面太软，则机器人可能卡住，运动受阻。建议使用商用地毯、瓷砖等材质。启智 ROS 机器人原则上在水平平面上工作，坡道坡度不大于 15 度，坡道倾斜度过大可能导致倾覆。

### 7.1.4 防护措施

启智 ROS 机器人不具备防水功能，在任何情况下，启智 ROS 机器人都不应该与雨水，雾，地面积水以及任何其他液体接触，否则可能导致电路和机构损坏。

### 7.1.5 温度和湿度

启智 ROS 机器人设计工作温度为 15° C 到 35° C 之间，使用中务必远离明火和其他热源。

### 7.1.6 电气概述

启智 ROS 机器人内部包含 1 个 USB-HUB、1 个启智电池模块、1 个电源控制板、1 个启智控制器、3 个启智伺服电机模块。位于机器人躯干部分外挂的计算机运行 ROS 操作系统，通过 USB 接口与机器人底盘内的 USB-HUB 连接。USB-HUB 将计算机的 USB 接口扩展为多路。扩展后的 USB 接口分别连接到启智控制器（以异步串口方式访问）、USB 转以太网接口、激光雷达、面板接口（用于用户自行连接设备，例如 U 盘、控制手柄）。

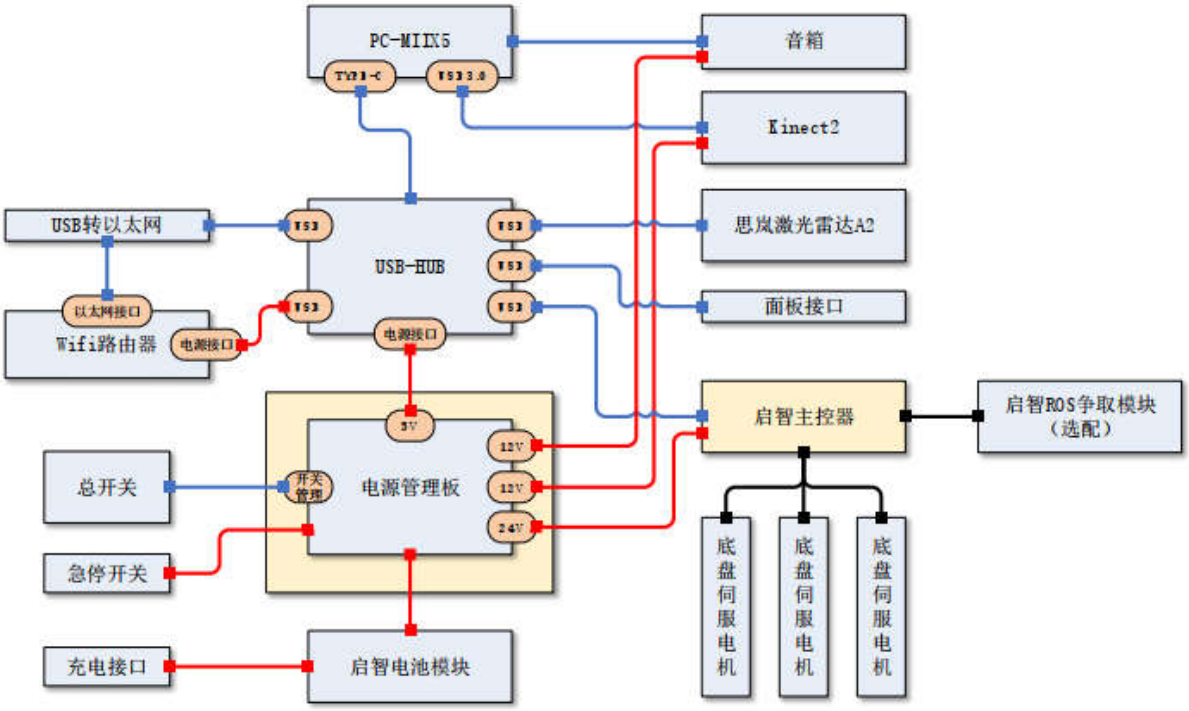



图 24 电气概述

图中红色线为供电，蓝色线为控制信号，黑色线为合并了供电于总线的专用线缆。

### 7.1.7 底盘控制器

启智控制器内部运行了启智 ROS 机器人专用固件，负责 PC 机于机器人之间的数据交互。PC 机将对底盘伺服电机的速度控制数据下发到启智控制器，由启

智控制器通过半双工 RS485 总线实时与 3 个底盘伺服电机（如果选配了抓取模块，则也包括抓取模块上的 2 个伺服电机）通讯。启智控制器同时接收伺服电机的反馈信息，解析出它们当前的位置、电流信息，与系统电压等信息汇总后发送到 PC 机。

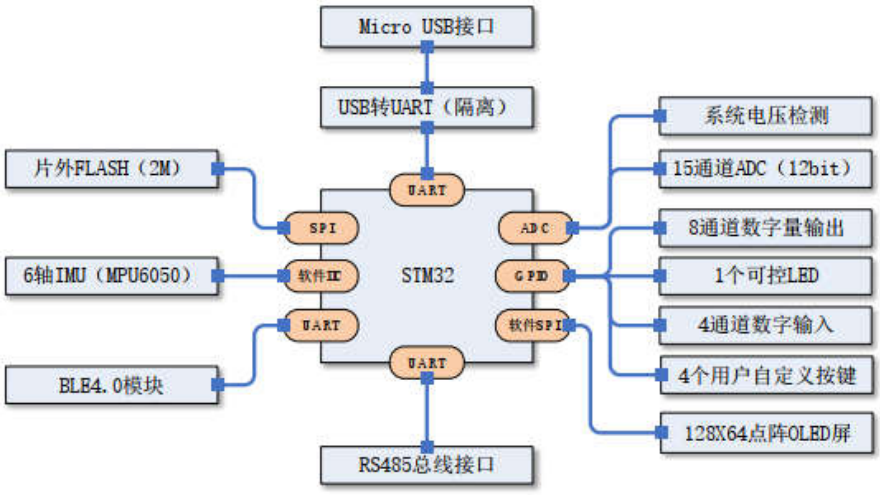



图 25 底盘控制

### 7.1.8 电源供电

启智 ROS 机器人的电源由电池模块供给，该电源模块内部由 7 枚 3500mA/h 容量的锂离子电池串联组成，内置电池保护板。该模块输出电压与当前剩余电量有关，剩余电量越少电压越低，正常工作输出电压范围 23.1V 至 29.4V。底盘内部的电源控制板控制开关机、执行机构的急停，并为各控制器、传感器提供电源。

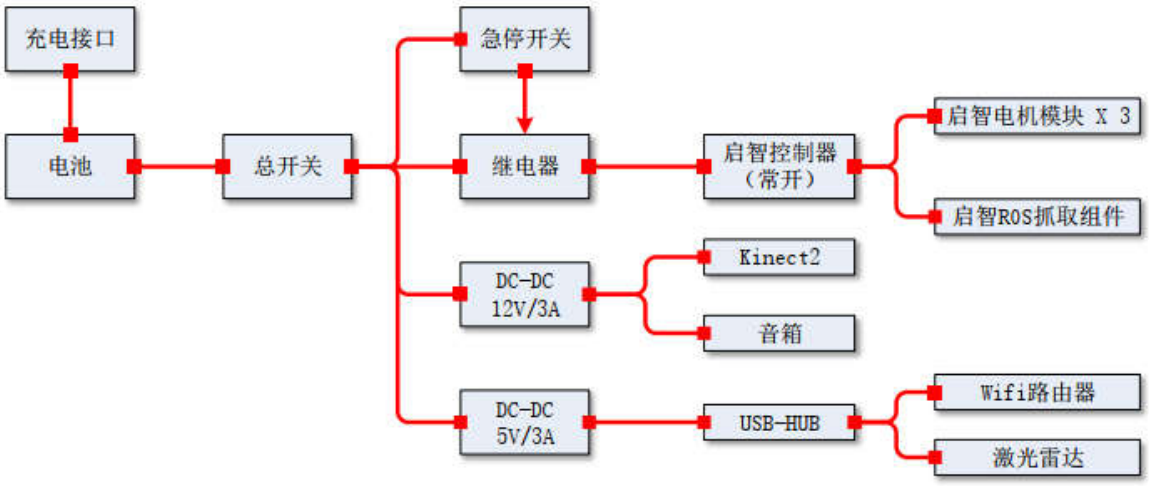



图 26 底盘供电

### 7.1.9 开关面板

面板上保留有总开关、急停开关、充电接口、2 个 USB 接口供用户使用。尾部为电池槽，可以快速更换电池模块。

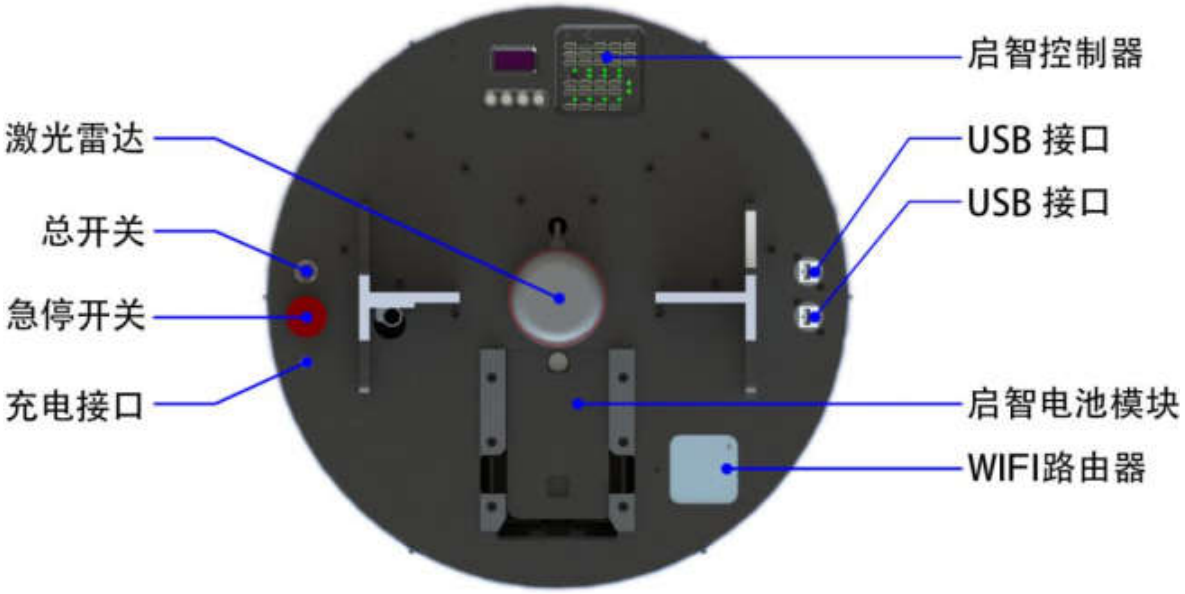



图 27 开关面板

### 7.1.10 通讯链路

启智控制器通过 RS485 总线与启智伺服电机模块通讯，以 50ms 为周期下发速度、位置等控制信息，并接收伺服电机反馈的实时位置和绕组电流信息。将系统内伺服电机模块的反馈数据进行重新封装后，整体以 50ms 周期向 PC 机发送。PC 机通过 USB 接口连接启智控制器，启智控制器内置 FTDI 接口转换芯片，把 USB 接口转换为 UART 串行接口，通过 UART 与启智控制器的控制核心 STM32 以 115200 波特率通讯。启智控制器在启智 ROS 机器人源码内以名为“wpb_home_core”的 node 作为通讯中枢。启智伺服电机模块内部包括有刷电机、正交编码器、行星齿减速器、驱动控制板。有刷电机的尾轴之间与正交编码器连接，输出轴与行星齿减速器连接，经减速后输出。启智伺服电机模块内置驱动控制板以 TI 的 TMS32F28062 数值信号控制器为核心，实现电流环、速度环、位置环控制。电流环伺服周期为 50us，速度环、位置环伺服周期为 1ms。

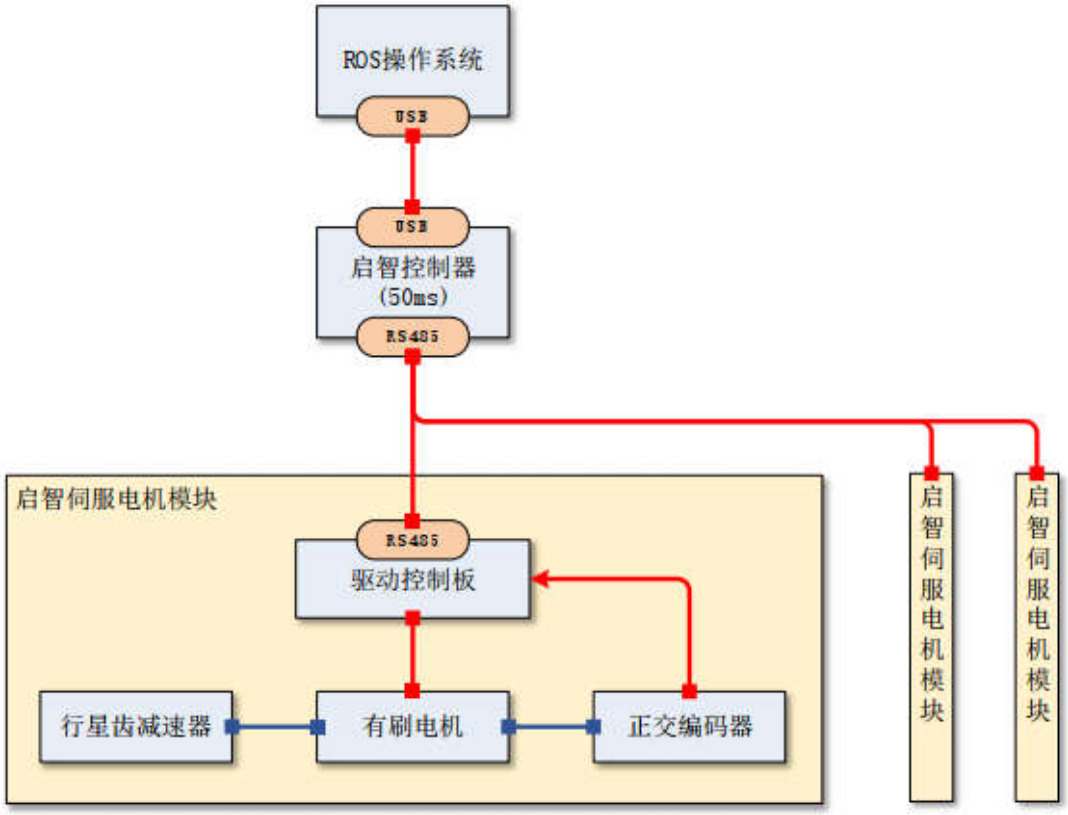



图 28 通讯链路

### 7.1.11 电机参数

表 2 电机参数

| | |
|---|---|
| 工作电压 | 24V |
| 额定功率 | 17W |
| 持续工作电流 | 1.4A |
| 空载转速 | 15000RPM |
| 减速比 | 64：1 |
| 编码器线数 | 12 线 |

在启智伺服电机模块内的驱动控制板算法中，将输入的速度信息进行了单位换算，输入速度为减速器输出轴的转速。模块反馈的位置信息为编码器反馈的位置信息：位置=（4×编码器线数）×电机转动圈数；即减速器输出轴旋转一周（机器人全向轮转动一周）对应位置为：（1×减速比）×（4×编码器线数）=64×48=3072。

### 7.1.12 传感器

表 3 传感器参数

| | |
|---|---|
| 激光雷达 | 思岚（SLAMTEC）RPLIDAR A2。测距范围：0.15 米-12 米；扫描角度：360°；测距分辨率：＜实际距离的 1%；角度分辨率：0.9°；扫描频率：10Hz。 |
| IMU | 启智控制器内置 MPU6050 惯性测量传感器 |

### 7.1.13 硬件结构

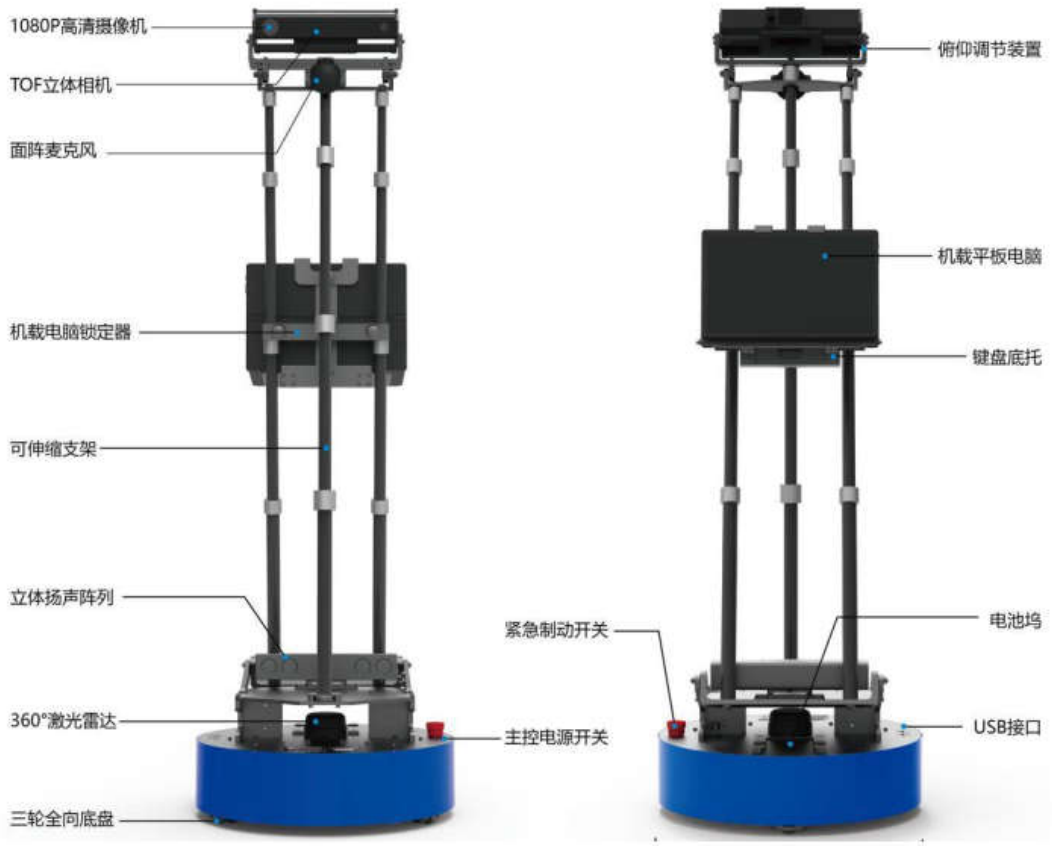



图 30 硬件结构

## 7.2 软件环境

### 7.2.1 Ubuntu14.04

Ubuntu（友帮拓、优般图、乌班图）是一个以桌面应用为主的开源 GNU/Linux 操作系统，Ubuntu 是基于 DebianGNU/Linux，支持 x86、amd64（即 x64）和 ppc 架构，由全球化的专业开发团队（Canonical Ltd）打造的。

其名称来自非洲南部祖鲁语或豪萨语的“ubuntu”一词，类似儒家“仁爱”的思想，意思是“人性”、“我的存在是因为大家的存在”，是非洲传统的一种价值观。

Ubuntu 基于 Debian 发行版和 GNOME 桌面环境，而从 11.04 版起，Ubuntu

发行版放弃了 Gnome 桌面环境，改为 Unity，与 Debian 的不同在于它每 6 个月会发布一个新版本。Ubuntu 的目标在于为一般用户提供一个最新的、同时又相当稳定的主要由自由软件构建而成的操作系统。Ubuntu 具有庞大的社区力量，用户可以方便地从社区获得帮助。Ubuntu 对 GNU/Linux 的普及特别是桌面普及作出了巨大贡献，由此使更多人共享开源的成果与精彩。

2013 年 1 月 3 日，Ubuntu 正式发布面向智能手机的移动操作系统。[4]

ubuntu 基于 linux 的免费开源桌面 PC 操作系统，十分契合英特尔的超极本定位，支持 x86、64 位和 ppc 架构。

2014 年 2 月 20 日，Canonical 公司于北京中关村皇冠假日酒店召开了 Ubuntu 智能手机发布会，正式宣布 Ubuntu 与国产手机厂商魅族合作推出 Ubuntu 版 MX3。魅族副总裁李楠到场出席。

与 Debian 稳健的升级策略不同，Ubuntu 每六个月便会发布一个新版，以便人们实时地获取和使用新软件。Ubuntu 共有五个长期支持版本(Long Term Support，LTS)：Ubuntu 6.06、8.04、10.04、12.04、14.04、16.04。Ubuntu 12.04 和 14.04 桌面版与服务器版都有 5 年支持周期。而之前的长期支持版本为桌面版 3 年，服务器版 5 年。每个 Ubuntu 的版本代号都是按照“形容词+动物”的格式命名的，一开始并不是按照字母顺序，从 6.06 的 Drapper DRAKE 才开始如此。而数字号则是表示发布的“年+月”，如 12.04 是在 2012 年 4 月发布。

本项目的开发和运行，都是基于 Ubuntu14.04 操作系统环境下开展的。

### 7.2.2 ROS

ROS（Robot Operating System，下文简称“ROS”）是一个适用于机器人的开源的元操作系统。它提供了操作系统应有的服务，包括硬件抽象，底层设备控制，常用函数的实现，进程间消息传递，以及包管理。它也提供用于获取、编译、编写、和跨计算机运行代码所需的工具和库函数。

ROS 的主要目标是为机器人研究和开发提供代码复用的支持。ROS 是一个分布式的进程（也就是“节点”）框架，这些进程被封装在易于被分享和发布的程序包和功能包中。ROS 也支持一种类似于代码储存库的联合系统，这个系统也可以实现工程的协作及发布。这个设计可以使一个工程的开发和实现从文件系统到用户接口完全独立决策（不受 ROS 限制）。同时，所有的工程都可以被 ROS

的基础工具整合在一起。

我们在这里所采用的 ROS 为 Indigo 版本的 ROS。Indigo 版本的 ROS 是目前用的人数比较多的一个版本，它对应 Ubunt14.04 版（13.10 也支持，不过单数版本比较少用就不说了），目前最新的 Kinetic 版本则对应 Ubuntu16.04。

### 7.2.3 Roboware Studio

我们的 IDE 采用的是 Roboware Studio。这是济南汤尼机器人科技有限公司基于 Visual StudioCode 开发的 ROS 专用 IDE。该软件有中文版本，还提供全中文的使用文档，十分适合中国的机器人开发者。

### 7.2.4 基础软件包

表 4 基础软件包

| Package 名称 | 内容 |
| --- | --- |
| wpb_home_bringup | 启智 ROS 机器人的基础功能 |
| wpb_home_behaviors | 启智 ROS 机器人的行为服务 |
| wpb_home_tutorials | 启智 ROS 机器人的应用例程 |
| wpbh_local_planner | 启智 ROS 机器人的导航局部规划器 |

启智机器人出厂标配的电脑里已经安装好 ROS、IDE 和启智机器人的源码包，可以直接使用。同时该软件包会在开源网站 Github 上持续进行维护更新。

### 7.2.5 扩展软件包

启智机器人除了基础功能的软件包以外，还提供了一个扩展软件包，该软件包里包含了大量复合任务的实现例程。因为该软件包需要依赖 xfyun_waterplus，iai_kinect，rplidar_ros，wpb_home 和 waterplus_map_tools 等第三方软件包，所以和基础软件包相互独立，以方便持续维护更新。

# 8. 需求可追踪性说明

## 8.1 功能需求可追踪性说明

### 8.1.1 启动和关闭机器人

本系统硬件部分包括机载电脑和机器人主体部分，由具有一定先验知识的超市工作人员将两部分通过 usb 接口正确连接，在保证机载电脑正常开机、机器人放置得当等前置条件的情况下，打开机器人硬件总开关并双击系统图标，即可开启系统。

系统开启后，主控制类 Control 状态机启动，将初始状态设置为等待“follow”指令，调度语音广播类 voidBroadcast 发出启动成功提示，并调度语音识别类 voiceDetect 开始监听，启动成功。

使用结束后，在确保机器人未处在执行抓取或建图任务中途的情况下，超市工作人员通过点击机载电脑系统界面关闭按钮，即可关闭系统，随即关闭机载电脑和机器人总开关，断开 usb 连接。

### 8.1.2 初始化地图场景建模

在 8.1.1 所述的开机状态下，voiceDetect 通过麦克风持续循环监听，一旦抓取到有效关键词，就生成相应指令返回给 Control。由于开机后机器人处于等待跟随指令状态，当 voiceDetect 抓取到有效关键词“follow”时，返回的指令内容为初始化建图，Control 随即会判断此指令可以被立即执行。接下来，Control 切换系统状态为建图，调度 voiceBroadcast 给工作人员播报反馈信息，调度地图建模类 Set_Map。Set_Map 中的 set 方法在调度导航类 Navigation 跟踪超市工作人员的过程中，对地图进行 SLAM 建图。

当机器人跟随超市工作人员回到超市入口处时，超市工作人员对麦克风发出停止跟随指令，voiceDetect 抓取到有效关键词“stop follow”，向 Control 返回停止建图指令。Control 判断状态可跳转，通知 Set_Map 调用 save 方法保存建好的地图文件并结束进程，通知 voiceBroadcast 给工作人员播报反馈信息，之后系统

状态切换为等待抓取指令。

### 8.1.3 实时地图查看

实时地图查看分为建图过程中和取物过程中两种情况。在前一种情况中，这一功能已经集成在 Set_Map 的 set 方法中。一旦系统进入建图状态，Set_Map 将实时显示当前地图临时文件，以地图界面作为 UI 界面的主体。在后一种情况中，Set_Map 已经完成场景建模，建模得到的地图文件通过 save 方法保存在数据库中。一旦系统状态切换为路径规划状态，Navigation 中的 route_plan 方法将调出保存的地图文件，同样以其作为 UI 界面主体，供用户实时查看。

### 8.1.4 关键物品地点记录

在 8.1.2 所述的建图状态下，voiceDetect 抓取到有效关键词“memorize+目标物”，返回记录指令给 Control。Control 转而提取出指令中的物品名称，移交给 Set_Map 进程，Set_Map 获取当前位置后，更新对应物品坐标为当前位置，并将以上过程完成情况反馈给 Control。Control 收到任务完成信息，调度 voiceBroadcast 给工作人员播报反馈信息，本次物品记录结束。

### 8.1.5 指定物品抓取

在 8.1.2 所述任务完成后，系统状态切换为等待抓取指令。此时，一旦 voiceDetect 抓取到有效关键词“目标物”，生成相应抓取指令返回给 Control。Control 判断当前可以执行抓取指令，切换系统状态为前往目标地，随即调度 voiceBroadcast 给用户播报反馈信息，调度 Navigation，使用 route_plan 方法开始规划前往目标物品的路线并记录顾客所在地。

到达指定地点后，Navigation 提示 Control 成功抵达，Control 将切换系统状态为开始抓取，调度 voiceBroadcast 播报反馈信息，调度目标检测类 TargetDetection 和抓取类 GrabObject 中的一系列方法，并在二者之间收集和传递必要信息。TargetDetection 和 GrabObject 将分别使用 CameraCollect 和 Arm 类与双目摄像头和机械臂发生关联。综合以上，完成物品识别和抓取过程。

同样，成功后，Control 会获得相应反馈信息，并通过 voiceBroadcast 向顾客

播报。Control 切换状态为返回顾客所在地，同样调度 Navigation 的 route_plan 方法完成此过程。成功返回后，Control 切换状态为移交物品，voiceBroadcast 通知顾客准备接收物品，调度 GrabObject 中的 release 方法将所抓取的物品放下。

### 8.1.6 关联错误条件

voiceDetect 从启动开始处于实时监听状态，一旦抓取到关键词即生成相应系统指令通知 Control，但该指令是否能够得到执行取决于当前的系统状态是否可以转移，这一判断交给 Control 来完成。对于可以立即执行的情况，Control 将按照 8.1.1~8.1.5 所述流程展开系列调度，对于不可执行的情况，Control 将不再执行实际动作，只通过调度 ExceptionHandler 向用户发出提醒。

每次启动时，Control 将首先检查系统状态并复位为初始态，以保证上次关机即使是非正常状态，也不会影响本次启动。

此外，各类在运行过程中发生意外情况，将以错误代码的形式返回给 Control，Control 调度 ExceptionHandler 分类处理异常。

## 8.2 非功能需求可追踪性说明

### 8.2.1 系统可靠性需求

我们的系统开发过程首先在 Gazebo 仿真环境中对每个用例和功能做到充分测试，同时能够存储测试过程各个模块的数据信息。保证系统通过测试后我们会使用实物进行再次测试。最终保证系统能够可靠地实现各种功能。

对于硬件的可靠性，我们可以根据硬件生产商提供的硬件检测软件或硬件来监控硬件的健康状况，可以通过顶起检修更换损坏的硬件。

### 8.2.2 系统可扩展性需求

系统使用面向对象开发方式，每个模块实现一部分功能，如果有新的功能，可以添加新的模块，并根据需要实现模块间的通信。

### 8.2.3 系统易用性需求

我们的系统需要管理员引导机器人构建地图信息，只需要管理员发出语音指令，并查看建图界面，来确认地图是否完成。

普通用户只要发出语音指令，即可让机器人实现对应功能，所以系统易用性很好。

### 8.2.4 系统安全性需求

系统可以记录日志文件信息，可以通过分析日志文件来判断系统执行过程中是否有数据被篡改，以此来判断系统是否按照预期执行。