智能抓取式机器人

软件设计说明书

SDD209

【版本号 1.0.0】

分工说明

小组名称	嵌不入队	
学号	姓名	本文档中主要承担的工作内容
17373548	吴天昊	3.3,4.1，6.9-6.14 节的编写，审核文章
17373545	聂志捷	4.2 节，第 5 部分的编写
17373513	吴仪周	3.2，4.3，4.5，6.15-6.18 节的编写
17373541	薛晨祺	3.1,4.4,6.1-6.8 节的编写
16061178	刘晓洁	第 1、2、7、8 部分的编写，并对文档进行整理

版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
1.0.0	4.19	刘晓洁	吴天昊	软件设计说明书第一版

目录

1 范围 1

1.1 项目概述 1

1.1.1 项目背景 1

1.1.2 主要功能 1

1.1.3 非功能性需求 2

1.1.4 应用场景 2

1.2 文档概述 2

1.3 术语和缩略词 3

1.4 引用文档 3

2. 需求概述 3

2.1 业务需求 4

2.1.1 普通家庭用户 4

2.1.2 管理员 4

2.2 数据需求 5

2.2.1 分析类 5

2.2.2 类的类型与关系 5

2.3 功能需求 5

2.3.1 机器人的主动控制 5

2.3.2 静态或动态障碍物避障 6

2.3.3 利用传感器实时建立环境地图 6

2.3.4 根据地图和自身位置实现动态路径规划和导航控制 6

2.3.5 检测、识别并定位环境中的特定目标，动态接近目标物 7

2.3.6 抓取目标物 7

2.3.7 语音交互 7

2.4 非功能需求 8

2.4.1 性能指标 8

2.4.2 质量属性 8

3. 数据库设计 9

3.1 名称：用户表 9
3.2 名称：物品表 9
3.3 名称：地图表 9
4. 体系结构设计 10
4.1 总体结构 10
4.2 软件体系结构 11
4.3 硬件体系结构 13
4.4 技术体系结构 14
4.5 支撑体系结构 15
5. 接口设计 17
5.1 系统用户界面（用户接口） 17
5.1.1 初始界面 17
5.1.2 选择操作页面 18
5.1.3 手动控制： 18
5.1.4 自动控制界面 19
5.1.5 弹出警告 19
5.2 外部接口 20
5.2.1 启智主控器 20
5.2.2 供电电源 21
5.2.3 激光雷达 21
5.2.4 Kinect2 相机 22
5.2.5 网络设备 22
5.3 内部接口 23
5.3.1 表现层/业务层接口 23
5.3.2 业务层/持久层接口 23
5.3.3 持久层/数据库层接口 23
6. 详细设计 24
6.1 User 类 24
6.2 Client 类 25

6.3 Administrator 类 25
6.4 Control 类 26
6.5 Advanced Control 类 27
6.6 Speech 类 28
6.7 Command 类 28
6.8 Debug 类 29
6.9 BasePlate 类 30
6.10 SlamMap 类 31
6.11 Position 类 32
6.12 Obstruction 类 33
6.13 Goal 类 33
6.14 DataSet 类 34
6.15 MechanicalArm 类 35
6.16 Navigation 类 36
6.17 Detect 类 37
6.18 Item 类 38
7. 运行与开发环境 39
7.1 运行环境 39
7.1.1 硬件环境： 39
7.1.2 硬件工作环境： 39
7.1.3 软件环境： 40
7.2 开发环境 40
7.2.1 硬件环境： 40
7.2.2 软件环境： 41
8. 需求可追踪性说明 41
8.1 功能需求 41
8.1.1 基本建图功能 41
8.1.2 机器人主动控制 41
8.1.3 导航功能 41

8.1.4 物品识别与抓取功能 42
8.1.5 语音交互 42
8.2 非功能需求 42
8.2.1 性能指标 42
8.2.2 质量属性 42

1 范围

1.1 项目概述

1.1.1 项目背景

现如今，智能服务已经占据了相当部分的市场，为此各商家研发设计了各种专用型机器人来为我们的生活提供便利。

而本项目的开发目的就是要设计制作一款具有包括抓取在内的多重功能的嵌入式机器人，以迎合各类市场的需求。嵌入式系统具有专用性强、实时性好、可靠性高的特点，并且还有较低的功耗，成本也非常低，因此对于我们开发机器人是一个非常好的选择。我们的开发平台基于启智 ROS 机器人，是一款为 ROS 机器人算法开发打造的硬件平台。我们的机器人将能够在一定范围内按照要求识别并抓取我们选定的目标，并且自动探测周围环境、实时建立地图并规划路径，同时还要具有躲避障碍物的功能，当然，为了进一步迎合用户需求，我们还会让机器人具备手动操纵以及语音控制的功能。

1.1.2 主要功能

本项目机器人将具备以下功能：

-单一类型目标的识别；

-多类型目标的识别；

-机器人的主动控制；

-利用传感器实时建立周边环境的地图；

-机器人导航过程中动静态避障；

-根据地图及目标点位置动态规划路径；

-对目标物按照姿态进行抓取；

-对异常情况进行识别与处理。

1.1.3 非功能性需求

我们对机器人的非功能性需求如下：

-低响应时间：我们希望系统对于用户的每个指令都能够快速反应，并且有较高的吞吐量和资源利用率；

-低功耗和较长的续航时间；

-可靠性：系统有较强的的鲁棒性，出错率较低，有一定的容错能力，并且机器人的整体结构也要足够牢固；

-易用性：用户界面美观，对用户使用较友好，易于操作；

-可扩展性：代码的编写按照模块化的进行，具有较高的复用性和可移植性；

-安全性：能够用户的权限进行区分，从而执行不同的功能，以免使用者出现误操作造成安全事故。

1.1.4 应用场景

家庭服务机器人：我们设计的机器人可以作为未来家庭服务机器人的一个重要功能模块，能够在用户家中实现精确导航避障，并且准确地对用户需要的目标进行抓取和传送。

分拣机器人：当前的快递点快递分拣的任务都是人工完成的，内容具有很大的重复性，而且人工还存在一定的犯错概率，我们设计的机器人可以用于对快递进行高效的分拣归类，并且出错率也可以保持在非常低的水平。

货运机器人：在商场中，当前有很多上货的任务都是人工进行的，很多时候货物重，难以分拣，我们就可以利用具有高强度抓取能力的机器人来进行搬运、上货的任务，既高效又节省成本。

1.2 文档概述

本文档是北京航空航天大学计算机学院2020年春季学期软件工程-嵌入式课程中的软件设计说明文档。本文档主要用于明确本次软件工程项目开发需要实现的功能和软件设计结构，小组成员包括吴天昊、聂志捷、吴仪周、薛晨祺和刘晓洁，其中吴天昊负责担任组长职务。在新冠肺炎疫情尚未结束之前，整个开发将

保持以线上沟通的方式来进行，具体的硬件实施将在后续依相关情况进行跟进。与本文档相关配套的，还有如下文档：SDP 软件开发计划文档、SRS 软件需求规格说明文档、STD 软件测试说明文档。本文档初次撰写于 2020 年 4 月 19 日。

说明书的内容按照项目概述，需求概述，数据库设计，体系结构设计，接口设计，详细设计，运行与开发环境，需求可追踪性说明进行组织和编写，每个部分详细阐述了项目的设计思路和框架，并填充了一些设计细节。

1.3 术语和缩略词

表 1-1 术语和缩略词

术语/缩略词	解释/全称
ROS	Robot Operating System/机器人操作系统
Gmapping	Gmapping 是基于滤波 SLAM 框架的常用开源 SLAM 算法，可以实时构建室内地图，在构建小场景地图所需的计算量较小且精度较高
Rviz	Rviz 是 ROS 官方提供的一款 3D 可视化工具，我们需要用到的所有机器人相关数据都可以在 Rviz 中展现
航点	含有明确坐标和编号的导航路径节点
地面点云	实时 SLAM 所探测到的周围障碍方位信息
基准格栅	地图文件中存储的障碍方位信息
行为脚本	针对某一关键词命令所设定的一串行动序列

1.4 引用文档

T. Wiedemeyer, “IAI Kinect2,” https://github.com/code-iai/iai_kinect2, Institute for Artificial Intelligence, University Bremen, 2014 – 2015, accessed June 12, 2015.

启智 ROS 机器人

启智 ROS 版_开发手册——20181109

2. 需求概述

2.1 业务需求

本项目旨在设计一个可语音操控、自动规划路径的抓捕型机器人，用户下载相应平台通过语音操作可以对机器人进行控制，完成物品的搬运工作。机器人能在复杂场景中进行环境建模，并进行路线规划，在行进过程中实现避障，最终查找并定位捕获所需物品。

2.1.1 普通家庭用户

1. 手动操控模式：

机器人的运动轨迹由用户进行操作，操作界面将提供给用户方向键进行操作，机器人只进行路障检测，当发现路障时进行警报并减速提醒用户，用户手动操作避过路障。当到达目的位置准备进行捕捉时，用户可从方向控制界面切换成目标捕捉界面，对机器人抓取臂进行操作，抓取到所需的目标物。

2. 自动模式：

用户使用语音或输入特定的物品与机器人进行交流，机器人在得到准确的物品指令后，会首先在自身存储的地图中寻找目标物，如果有将直接进行路径规划去寻找；如果没有，将从目前位置进行 360 度扫描进行建模和目标物的确定，再实行查找捕捉。在捕捉完成之后，搜索用户的位置，选择路径把物品交给用户。在此过程中，机器人将有避障、建模、路径选择的能力；用户可以随时语音或者手动选择停止机器人的抓捕任务。

2.1.2 管理员

可以对于机器人后台的路径进行进一步规划和调整，对于普通家庭用户的权限进行控制。在登录管理者界面后，可对机器人的基本情况进行操作，如清空后台数据，管理增加和删除普通用户信息等，同时管理者账号也可以对机器人进行如普通用户一样的操作。

2.2 数据需求

2.2.1 分析类

表 2-1 分析类筛选表

潜在类	表现形式
普通家庭用户	角色
管理者	角色
机器人	事物
控制界面	事物
用户	角色
路障	事物
航点	事物
抓取臂	事物
目标物	事物

2.2.2 类的类型与关系

实体类：普通用户、管理员、用户、机器人、机械臂、物体、航点、障碍物

边界类：移动控制、语音交互

控制类：用户控制、管理员控制、异常处理

2.3 功能需求

2.3.1 机器人的主动控制

本功能的实现依靠 ros.h 与速度结构体类型 geometry msg::Twist 的定义文件实现。在打开机器人底盘面板的红色急停开关后，可以采用以下几种方式控制机器人的移动：

（1）前后左右移动：手柄的左侧摇杆，控制机器人运动的平移向量，上下分别控制前后移动，左右分别控制侧向平移；

（2）旋转：手柄的右侧摇杆，控制机器人运动的转动向量，拨向左是左转，拨向右是右转；

（3）旋转移动：机器人的运动平移向量和转动向量可以叠加的，即可以一边平移一边旋转。

2.3.2 静态或动态障碍物避障

本功能的实现依靠机器人所带的激光雷达实现。在开启激光雷达后，可在Rviz界面看到周围360°的障碍物分布情况，并形成障碍物轮廓的俯视二维点阵输入到ROS系统中。静态或动态障碍物避障功能可实现：

（1）监测：在移动过程中实时监测运动路径上的障碍物和距离；

（2）停止：在距离障碍物一定距离时，停止运动

（3）判断：选择左侧或右侧中更空旷的一侧，并将机身进行旋转；

（4）躲避：向当前方向前进一定距离，直至原本行驶方向路径上无障碍物；

（5）恢复：停止平移并旋转回原本行驶方向，继续移动。

2.3.3 利用传感器实时建立环境地图

本功能的实现依靠Gmapping算法技术实现。运行Gmapping算法期间可以观察到以下情况：

（1）扫描：机器人周围的地面基准变成了深灰色，出现白色扫描区域；

（2）识别：探测到的障碍点在Rviz上呈红色；静态障碍物轮廓呈黑色；

（3）更新：移动机器人时，白色区域增加，静态障碍物的轮廓越来越接近；

（4）存储：绕场地移动一圈后，可以得到保存的后缀为.pgm的地图文件。

2.3.4 根据地图和自身位置实现动态路径规划和导航控制

本功能的实现主要依靠蒙特卡洛自适应定位算法、move_base软件包和MapTools软件。在调节优化至红色激光雷达数据点和静态障碍物轮廓大致贴合后，可以通过以下步骤进行导航控制：

（1）在Rviz视图中加入若干个目标位置的航点以及机器人的朝向；

（2）机器人自动规划路径并按照顺序依次移动至航点；

（3）到达一个航点后，机器人不停止，立即规划至下一个航点；

（4）在抵达最后一个航点后，机器人停止移动，并旋转至设定的朝向；

在导航期间可以实现的其他功能包括：

（5）可在代码中修改航点坐标或增减航点数目，机器人将重新规划路径；

（6）对路径上的障碍物可以有效躲避，不发生碰撞；

2.3.5 检测、识别并定位环境中的特定目标，动态接近目标物

本功能的实现主要依靠 Kinect2 视觉传感器、PCL 平面检测算法。在对 Kinect2 的角度倾角参数进行标定、调节地面点云和基准栅格完全重合后，可以实现以下功能：

（1） 接近：根据导航指令接近桌子或货架、机器人正面面向桌子或货架；

（2） 检测：将桌面上的物品标注出来，并计算其三维坐标；

（3） 识别：机器人根据指令选定目标物品；

（4） 动态调整：机器人调整自己与桌面的距离并左右平移对准物品。

2.3.6 抓取目标物

本功能的实现依靠 ros.h 与机器臂结构体类型 sensor_msg::JointState 的定义文件实现。在正确安装机器臂的前提下，在目标物正前方，能按照如下顺序使机械臂状态切换：

（1）初始：手臂处于最下端折叠收起状态；

（2）展开：手臂从零位上升到展开状态的最低位；

（3）上升：机器臂上升指定高度，直至物体中心位置高度；

（4）抓取：机械臂调节手抓的两指间距，直至夹住物体

（5）提起：机器臂继续上升制定高度，直至物体脱离桌面，悬停空中

2.3.7 语音交互

本功能的实现依靠科大讯飞的语音识别引擎和阵列麦克风实现。主要能够进行以下涉及语音交互的行为：

（1）设置：自定义目标关键词，并可以配置关键词所对应的行为脚本；

（2）识别：用户对着麦克风说出目标关键词，可准确识别语音命令结果；

（3）执行：机器人根据行为脚本无差错执行命令；

（4）待命：在完成命令后进入监听状态，等待下一条命令输入。

2.4 非功能需求

2.4.1 性能指标

- 响应时间：在95%的情况下,能在1s内对一般指令做出反应，能在5~10s内对语音指令做出反应。
- 功耗：机器人电源由电池模块供给,工作电压24V,额定功率14W,持续工作电流1.4A,可以持续工作4小时以上。
- 处理能力要求：机器人内部的信息具有较高的更新率和反馈率；能在较短时间内对用户期望的目标进行识别和反应，快速避障；具有较高的精准度,能够准确移动到目标位置,准确识别,抓取物体,误差<=0.1m。

2.4.2 质量属性

- 可用性：控制指令简单，易于用户操作；具有一定容错能力；提供数据备份和恢复功能。
- 安全性：区分用户权限，实现不同的功能；记录运行日志，记录运行中的系统错误和用户关键操作信息。
- 可移植性：仅支持Ubuntu16.04及ROS Kinetic环境下运行，暂不支持移植。
- 可维护性及可扩展性：保留历史各版本的源代码；系统功能模块化；限定修改和bug修复时间。有清晰的系统结构，规范。

3. 数据库设计

3.1 名称：用户表

标识：User

用户表包括：用户名、密码、权限三个字段。

表 3-1 数据库用户表

名称	字段名称	数据类型	主键	非空
用户名	name	文本	Yes	Yes
密码	password	文本	No	Yes
权限	isAdmin	文本	No	Yes

3.2 名称：物品表

标识：Item

物品表包括：物品名、三维坐标、物品形状、物品标签四个字段。

表 3-2 数据库物品表

名称	字段名称	数据类型	主键	非空
物品名	name	文本	Yes	Yes
三维坐标	position	文本	No	Yes
物品形状	box	图像	No	Yes
物品标签	text	标号	No	Yes

3.3 名称：地图表

标识：Map

地图表包括 label、x_position、y_position 和 type 四个字段

表 3-3 数据库地图表

名称	字段名称	数据类型	主键	非空
标签	label	str	Yes	No

横坐标	x_position	double	No	No
纵坐标	y_position	double	No	No
类型	type	str	No	No

4. 体系结构设计

4.1 总体结构

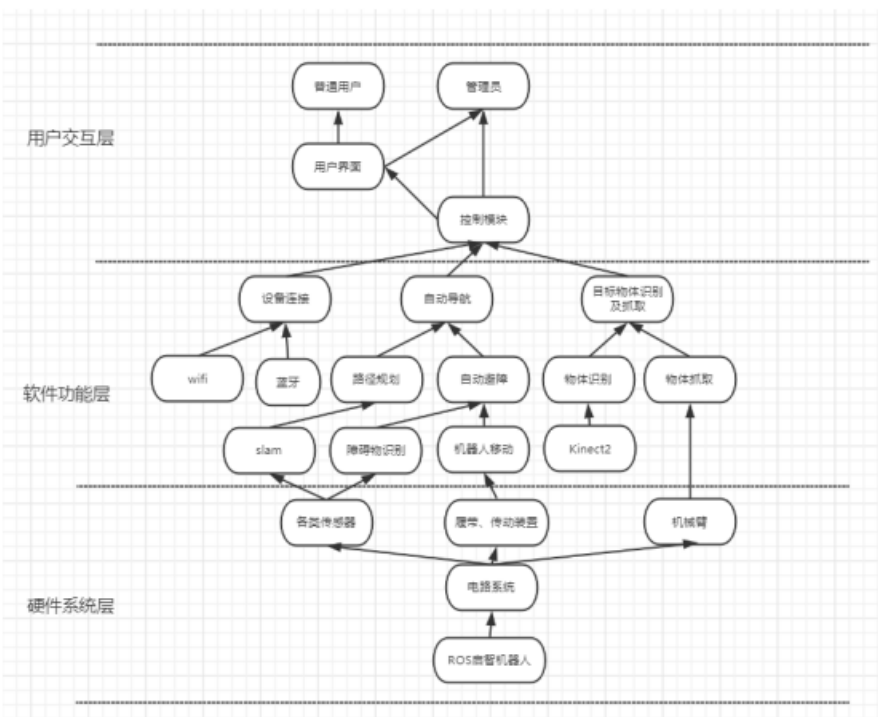


图 4-1 总体结构图

整体上，在用户交互层，我们希望使用者通过用户界面来与机器人进行交互，与此同时，管理员既可以使用用户界面来控制系统，也可以直接对控制模块下的每个环节进行统一管理。向下我们将机器人的软件功能层整体分为设备连接、自动导航和目标物体识别抓取三个部分。设备连接部分包括 WIFI 和蓝牙两种连接方式。而自动导航则分为路径规划和自动避障两个环节，二者各自由 slam 和障碍物识别作为基础，并且用户还可以选择是否手动控制机器人的行进路线。第三

部分则是基于 Kinect2 的物体识别以及物体的抓取，我们将针对物体的不同形状来调整机械臂的姿态从而获得最好的抓取效果。同时，整个软件功能层都建立在硬件系统上，包括各类传感器、传动装置和机械臂，以及整体的电路系统（电源、电机和电线）。将以上这三个层次组合在一起就构成了我们整体的系统。

4.2 软件体系结构

图 4-2 软件结构图

在软件结构上，本系统使用含隔离层的分层结构。用户通过语音输入来命令机器人进行提前录入的一系列指定行动序列，通过浏览器上的 GUI 可对导航、目标检测等功能进行交互操作，还可以通过手柄控制来直接操控机器人的移动。而用户的需求通过层与层之间的接口进行传递，直到需求被满足，按照输入的路径将结果重新传递为表现层。

若用户需求中设计到对数据库某一项表中元素的增删改查，如导航到预定地点或删除用户账号，则需求将逐层传递给数据库，以完成对数据库特定表项的读写；若用户需求在持久层便可得到满足，则需求则在持久层实现需求后之间返回。特殊情况时，当系统由于各种原因发生异常中断时，异常信息与相关参数也将自动保存在数据库中，方便编程人员分析研判。

由于传统的封闭架构层次导致用户通过手柄进行一些简单的功能时（如机器

人移动等），需求需要通过服务层才能访问到持久层，因此我们选择将服务层开放，这样部分请求可以绕过这一层而直接访问下面的层。

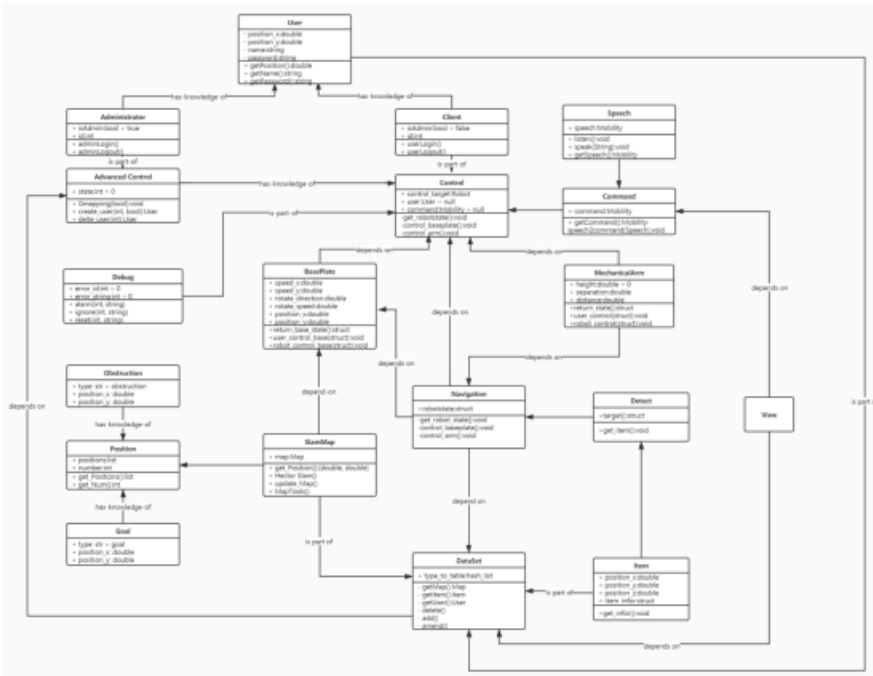


图 4-3 UML 类图

表 4-1 设计类表

类型	类名	继承类
实体类	User	
实体类	Client	User
实体类	Administrator	User
实体类	Position	
实体类	Obstruction	Position
实体类	Goal	Position
实体类	Item	
实体类	Command	
实体类	DataSet	
控制类	Control	

控制类	AdvancedControl	Control
控制类	BasePlate	
控制类	SlamMap	
控制类	Navigation	
控制类	Detect	
边界类	Speech	
边界类	View	
边界类	Debug	

4.3 硬件体系结构

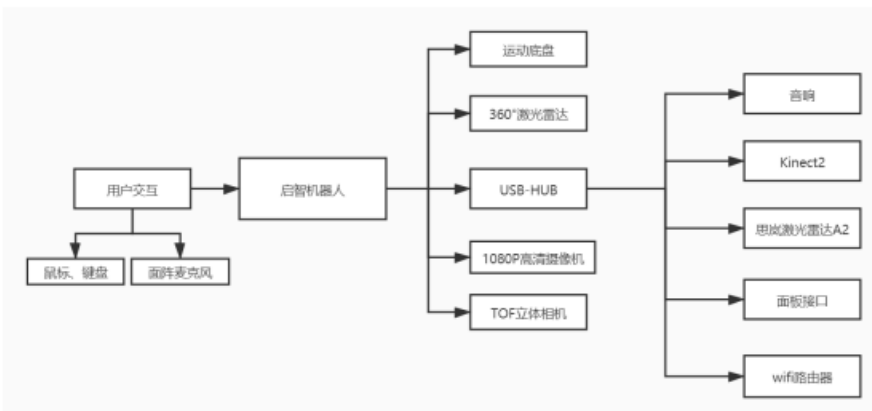


图 4-4 硬件体系结构

此机器人所涉及的硬件主要包括鼠标、键盘、麦克风等用于用户交互的硬件，以及启智机器人所包含的用于其捕捉信息和行进运动的硬件，主要有运动底盘、360° 激光雷达、1080P 高清摄像机、TOF 立体相机和 USB-HUB 所拓展的一些设备，如：音响，Kinect2，思岚激光雷达 A2，WIFI 路由器，面板接口等。这些硬件设备将支持此服务型机器人的运行。

4.4 技术体系结构

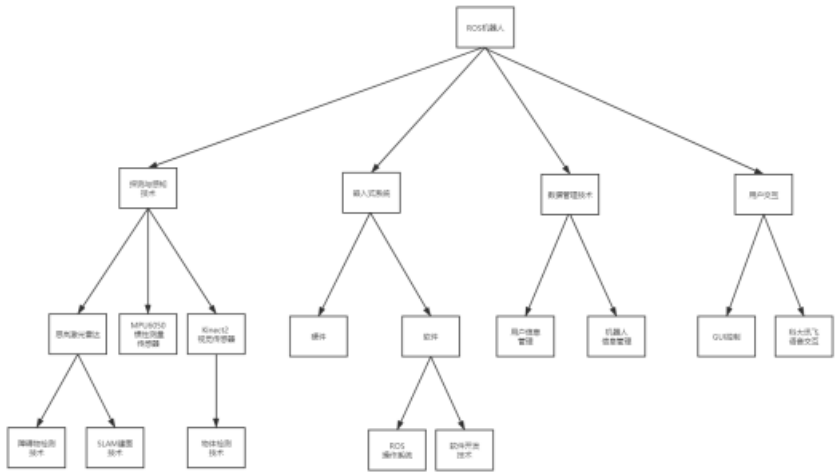


图 4-5 技术体系结构

技术体系结构定义了整个信息系统中的技术环境和基础结构，分为以下四部分：

(1) 探测与感知技术：利用各类传感器对机器人周围环境进行探测，将环境信息转换为机器人可用的数据，用于建图、检测物体、避障等功能。

(2) 嵌入式系统：开发人员进行嵌入式系统的开发所需的软硬件技术，包括驱动机器人运动、机械臂的控制、软件包开发、接口设计等技术。

(3) 数据管理技术：利用数据库等技术，对机器人获取的数据和用户信息进行存储和管理。

(4) 用户交互：实现人机交互，主要为 GUI 的设计，以及通过语音识别功能实现语音交互。因为语音交互需要使用科大讯飞的在线语音识别，还需要 WIFI 技术来支持联网。

4.5 支撑体系结构

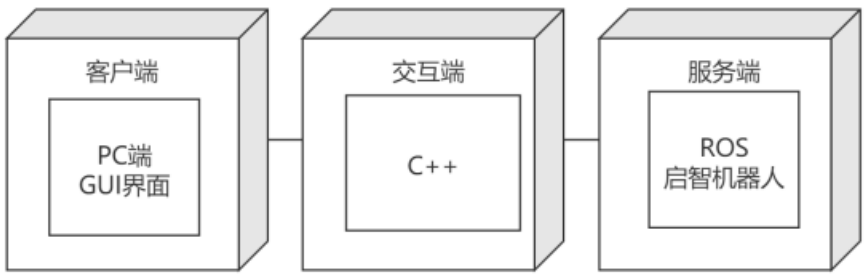


图 4-6 系统部署图

本产品使用 ROS 启智机器人进行物体的运送和抓取，用户只需要在自己的 PC 端使用相应的 GUI 界面进行控制，即可对于机器人进行操控。本产品后台的使用 C++，使其具有较快的相应速度和交互能力。

项目阶段	开始	结束
实现机器人控制	2020/3/9	2020/3/13
机器人的静态障碍物避障	2020/3/14	2020/3/18
机器人的动态障碍物避障	2020/3/16	2020/3/20
建立周围环境地图	2020/3/21	2020/3/24
实时动态路径规划	2020/3/25	2020/3/31
机器人导航控制	2020/4/1	2020/4/5
检测、识别、定位单一目标	2020/4/1	2020/4/5
接近目标物，实时更新路线	2020/4/6	2020/4/10

	开始	结束
抓取目标物	2020/4/6	2020/4/15
检测、识别、定位环境中多个目标	2020/4/13	2020/4/30
实现机器人语音交互	2020/4/15	2020/4/28
撰写各项目报告	2020/3/9	2020/5/10
准备展示答辩	2020/3/9	2020/5/10
机器人各项功能检验	2020/4/28	2020/5/10

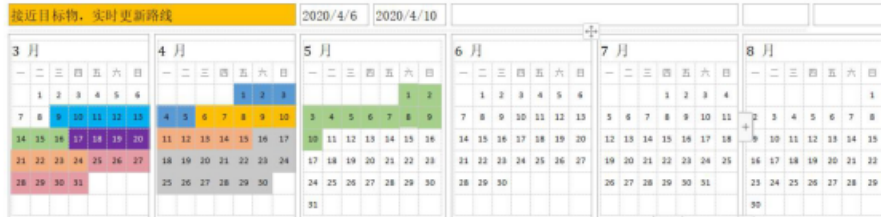


图 4-7 实施方案图

项目阶段	人员分工
实现机器人控制	吴天昊、聂志捷、吴仪周
机器人的静态障碍物避障	吴仪周、薛晨祺、刘晓洁
机器人的动态障碍物避障	吴天昊、聂志捷、薛晨祺
建立周围环境地图	吴仪周、薛晨祺、刘晓洁
实时动态路径规划	聂志捷、薛晨祺、吴仪周
机器人导航控制	吴天昊、吴仪周、聂志捷
检测、识别、定位单一目标	薛晨祺、刘晓洁、吴天昊
接近目标物，实时更新路线	吴仪周、吴天昊、薛晨祺
抓取目标物	聂志捷、吴天昊、刘晓洁
检测、识别、定位环境中多个目标	吴仪周、聂志捷、薛晨祺
实现机器人语音交互	刘晓洁、吴仪周、吴天昊
撰写各项目报告	吴天昊、聂志捷、吴仪周、薛晨祺、刘晓洁
准备展示答辩	吴天昊、聂志捷、吴仪周、薛晨祺、刘晓洁
机器人各项功能检验	吴天昊、聂志捷、吴仪周、薛晨祺、刘晓洁

图 4-8 人员分工

本项目有详细的实施方案和人员分工计划，将尽可能保证产品的时效以及质量。同时，在开发过程中，采用迭代-增量模式，将一步步完善和落实用户所需功能。在这之中，增加对于代码的测试强度，完善系统的使用体验。

5. 接口设计

5.1 系统用户界面（用户接口）

5.1.1 初始界面

图 5-1 初始界面

初始界面主要包括登录和注册界面，用户可以登录和注册，系统将自动识别是普通用户还是管理员。

5.1.2 选择操作页面

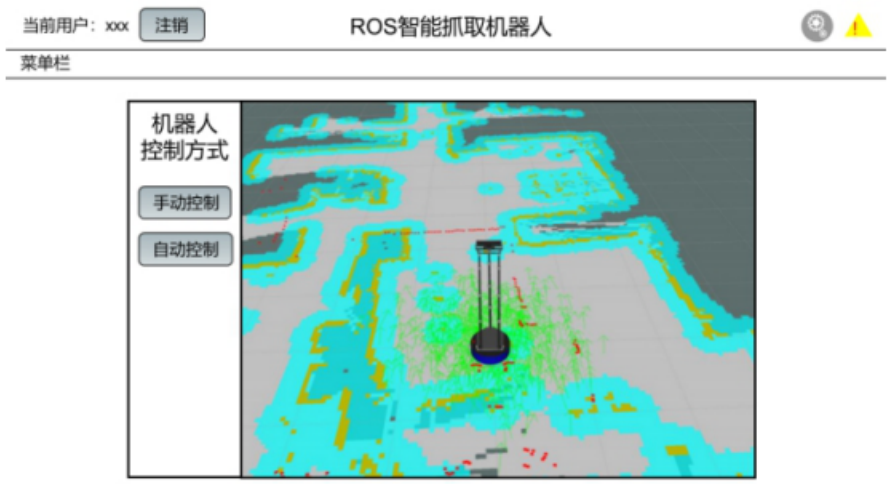


图 5-2 选择操作界面

用户登录后会在左上角显示当前用户名和注销键，同时将显示机器人目前所在位置。用户可根据需要选择手动控制或者自动控制。

5.1.3 手动控制：

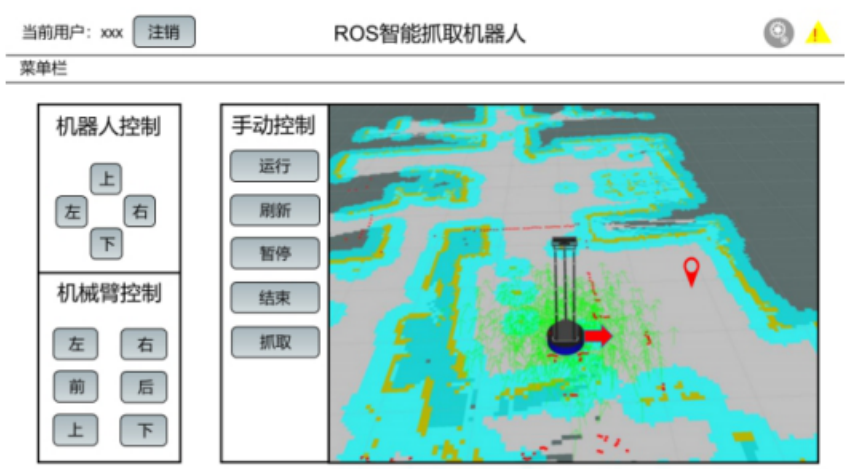


图 5-3 手动控制界面

用户可依照左边的按钮对于机器人进行手动控制，当调整到合适位置按抓取按钮进行抓取。同时，还有暂停，结束等功能。

5.1.4 自动控制界面

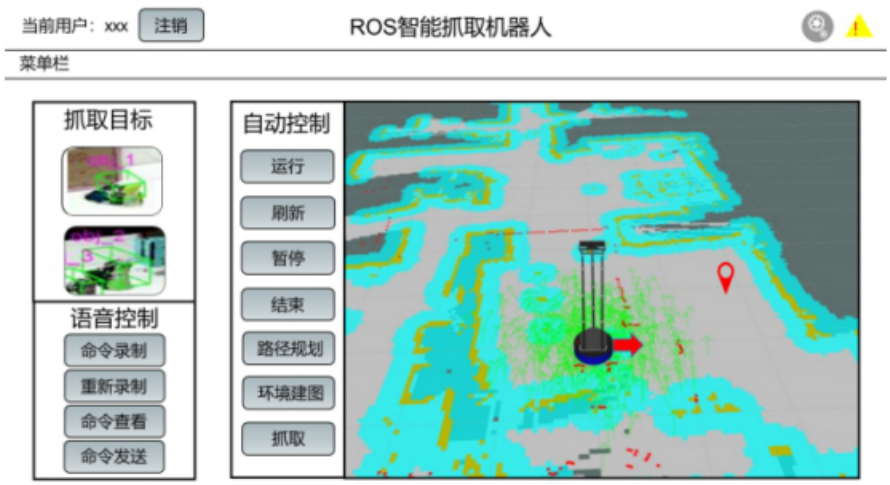


图 5-4 自动控制界面

用户选择自动控制时，与机器人进行语音交互，机器人识别语音后会在左上方显示目标抓取物的示例，用户可依此进行判断是否为自己所需抓取的物品，右侧包括暂停、结束等功能按键。

5.1.5 弹出警告

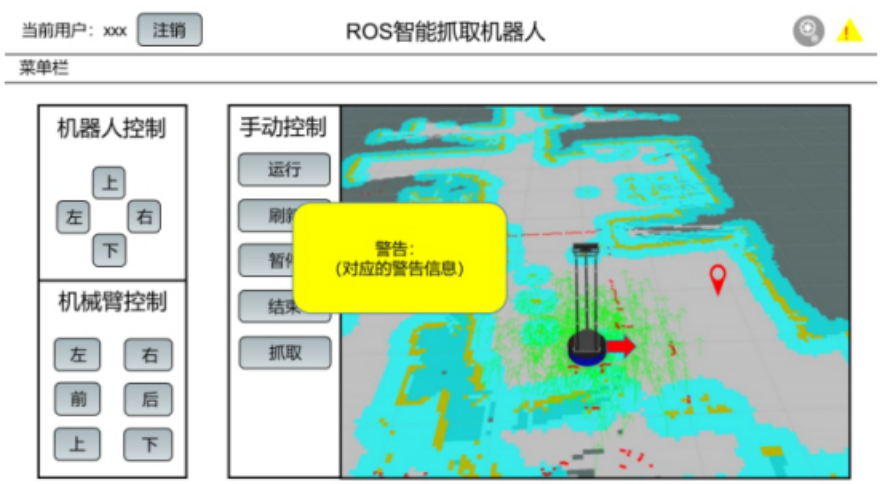


图 5-5 警告样例

在手动控制过程中，当机器人遇到障碍物时会在页面提示警告，同时报警减速。

5.2 外部接口

计算机外挂在位于机器人躯干部分的支架上，运行 ROS 操作系统（Ubuntu14.04 支持版本）。机器人底盘上的 USB-HUB 与计算机的 USB 端口连接。通过 USB-HUB 将信号传递给多路，分别连接启智控制器（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，如 U 盘或控制手柄）

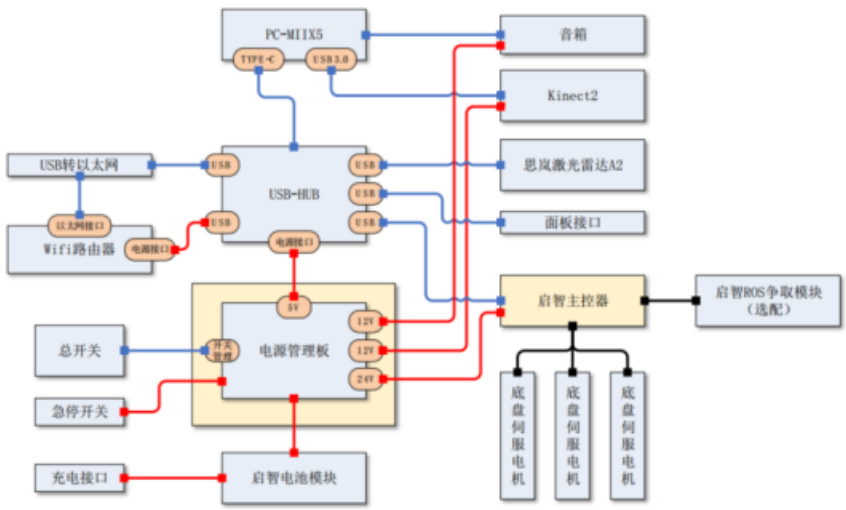


图 5- 6 供电与总控模块

5.2.1 启智主控器

启智底盘控制器内部运行启智 ROS 机器人专用固件，负责 PC 与机器人之间的数据传输。具体的发送数据、链路传输形式和数据接口如下表所示：

表 5- 1 启智主控器数据传输参数表

发送数据	链路形式	数据接口
速度控制数据	半双工 RS485	3 个底盘伺服电机
抓取控制数据	半双工 RS485	2 个机械臂伺服电机
伺服电机反馈数据 （位置、电流与电压等信息）	Micro USB	PC 机

5.2.2 供电电源

启智机器人的电源由电源模块供给。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，详细电源模块参数由下表所示：

表 5- 2 电池模块参数表

电池类型	充电式锂电池
电池容量	3500mA/h
电池数目	7 枚
附加信息	串联；内置电池保护板
正常工作电压	23.1V-29.4V

为各控制器、传感器提供电源，参数如下：

表 5- 3 电池供电参数表

供电参数	供电对象
急停开关	继电器
继电器	启智控制器
DC-DC 12V/3A	Kinect2
	音响
DC-DC 5V-3A	Wifi 路由器
	激光雷达

5.2.3 激光雷达

激光雷达是地面移动机器人的常用传感器，使用高速旋转的激光测距探头将周围 360° 的障碍物分布状况测量出来，形成障碍物轮廓的俯视二维点阵输入到 ROS 系统中。机器人携带的激光雷达由 USB-HUB 供电，相关数据通过 USB 接口传送给 PC，其详细参数如下表所示：

表 5- 4 激光雷达参数

型号	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

5.2.4 Kinect2 相机

Kinect2 相机具有普通镜头、Tof 镜头和红外传感器，能够探测机器人前端一定距离的物体三维坐标。相机直接由电源模块进行独立供电，数据传输通过 USB3.0 端口与 PC 相连接，其详细参数如下表所示：

表 5- 5 Kinect2 相机参数

色彩	分辨率	1920×1080（32bit）
	帧率	30/15fps
深度	分辨率	512×424（16bit）
	帧率	30fps
红外相机	分辨率	512×484
	频率	30Hz
可检测范围	0.5-4.5m	
深度误差	＜检测范围的 0.5%	
角度	水平	70°
	垂直	60°

5.2.5 网络设备

启智机器人具有 wifi 路由器以及以太网接口，可以支持有线 /无线的数据传输。其中 wifi 路由器由 USB-HUB 通过 USB 进行供电；以太网转换模块使用以太网端口与 wifi 路由器连接，使用 USB 转接头与 USB-HUB 进行连接。

5.3 内部接口

由于本模型使用了层次模型作为软件体系结构，因此只需要实现相邻层之间交互的接口即可。每个用户需求都被逐层分解成子任务，而每个子任务都处于一

个特定的抽象级别，要么在本层执行完毕，要么继续分解并将指令传输给下一层。

5.3.1 表现层/业务层接口

表现层/业务层接口主要是将用户的需求转化为功能模块对应的指令；对于下层呈递的用户数据，以特定的格式展示在前端。因此对应的内部接口有语音识别接口、数据格式映射接口。

5.3.2 业务层/持久层接口

业务层/持久层接口主要是将业务层的数据需求交付给持久层；将业务逻辑所需数据从持久层传递给业务层。此外，持久层还需要将命令执行过程中的异常信息反馈给业务层，业务层进行相应的异常中断处理，因此对应的内部接口有SLAM 地图绘制接口、物体三维坐标扫描接口、导航路径点读取接口、底盘与机器臂运动控制接口和异常中断报告接口。

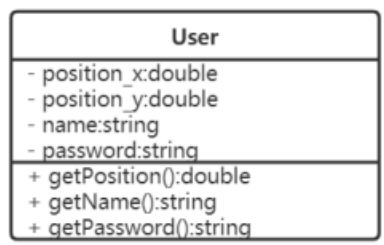
5.3.3 持久层/数据库层接口

持久层/数据库层接口主要是将对数据库中存储的特定信息进行增删改查的工作，在数据库设计中包括用户信息、Kinect 检测的物体信息以及地图坐标系下的路径点与障碍物位置信息。因此对应的内部接口有用户表读/写接口、物体表读/写接口和地图表读写接口。

6. 详细设计

6.1 User类

6.1.1 类图



User 类指用户的相关信息，包括普通用户和管理员。

6.1.2 属性信息

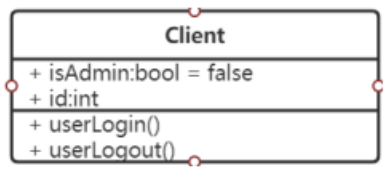
属性名	类型	说明
position_x	double	机器人识别的用户 x 坐标
position_y	double	机器人识别的用户 y 坐标
name	string	用于登录的用户名
password	string	用于登录的密码

6.1.3 方法信息

方法名	说明
getPosition():double	获取用户当前位置
getName():string	获取用户名
getpassword():string	获取用户密码

6.2 Client 类

6.2.1 类图



Client 继承了 User 类,是普通用户,可以对机器人进行控制。

6.2.2 属性信息

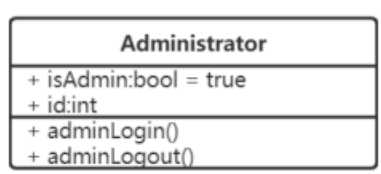
属性名	类型	说明
isAdmin	bool	初始值为 flase,表示该用户为普通用户
id	int	用户的唯一 id

6.2.3 方法信息

方法名	说明
userLogin()	用户登录
userLogout()	用户登出

6.3 Administrator 类

6.3.1 类图



Administrator 继承了 User 类,是管理员,可以对机器人进行更高级的控制。

6.3.2 属性信息

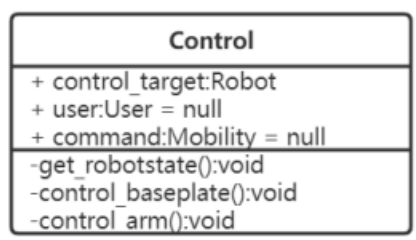
属性名	类型	说明
isAdmin	bool	初始值为 true,表示该用户为管理员
id	int	管理员的唯一 id

6.3.3 方法信息

方法名	说明
adminLogin()	管理员登录
adminLogout()	管理员登出

6.4 Control 类

6.4.1 类图



Control 类是对机器人的行动进行控制,普通用户和管理员都拥有控制权限。

6.4.2 属性信息

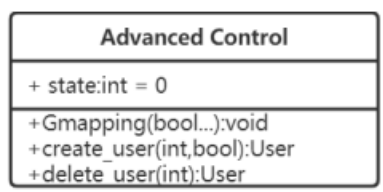
属性名	类型	说明
control_target	Robot	所控制的机器人类
user	User	已登录的用户
command	Mobolity	机器人控制指令流

6.4.3 方法信息

方法名	说明
get_rebotstate():void	获取机器人当前状态
control_baseplate():void	对机器人底盘进行控制
control_arm():void	对机械臂进行控制

6.5 Advanced Control 类

6.5.1 类图



Advanced Control 类是对机器人的高级控制类，只有管理员拥有权限。

6.5.2 属性信息

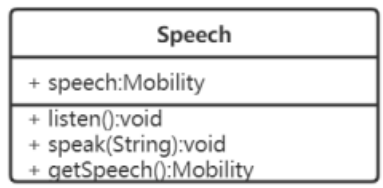
属性名	类型	说明
state	int	机器人的状态，初始值为 0

6.5.3 方法信息

方法名	说明
Gmapping(bool...):void	管理员进行建图
create_user(int,bool):User	管理员创建新用户
delete_user(int):User	管理员删除已有用户

6.6 Speech类

6.6.1 类图



Speech类进行语音交互的处理。

6.6.2 属性信息

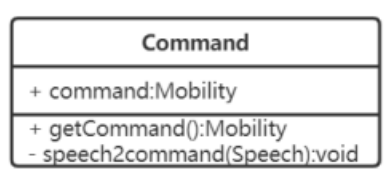
属性名	类型	说明
speech	Mobility	机器人接受的语音流

6.6.3 方法信息

方法名	说明
listen():void	机器人接受语音指令
speak(String):void	将字符串转变为机器人要发出的语音
getSpeech():Mobility	获取语音流

6.7 Command类

6.7.1 类图



Command类管理控制机器人运行的指令。

6.7.2 属性信息

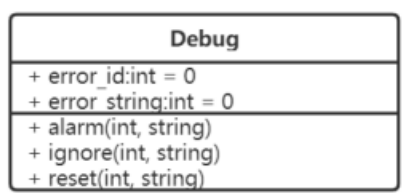
属性名	类型	说明
command	Mobility	机器人控制指令流

6.7.3 方法信息

方法名	说明
getCommand	获取指令流
speech2command	将语音指令转换为当前指令类型

6.8 Debug 类

6.8.1 类图



Debug 类对机器人运行过程中的错误进行处理。

6.8.2 属性信息

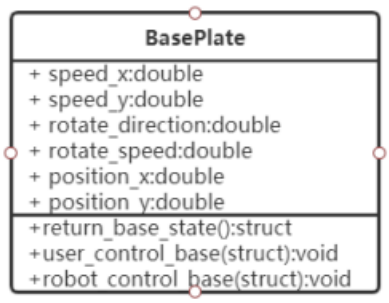
属性名	类型	说明
error_id	int	错误类型
error_string	string	错误信息

6.8.3 方法信息

alarm(int,string)	发出警报提醒用户
ignore(int,string)	忽视当前错误
reset(int,string)	重置机器人状态

6.9 BasePlate类

6.9.1 类图



BasePlate类对机器人的底盘运动和反馈进行管理。

6.9.2 属性信息

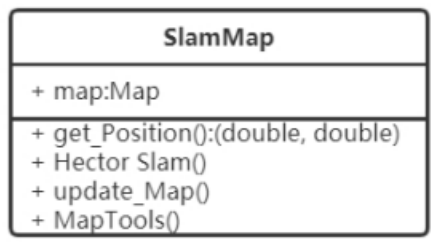
属性名	类型	说明
speed_x	double	机器人底盘在x轴方向的速度
speed_y	double	机器人底盘在y轴方向的速度
rotate_direction	double	机器人底盘旋转的方向（逆时针/顺时针）
rotate_speed	double	机器人底盘旋转的角速度
position_x	double	机器人在空间中的x坐标
position_y	double	机器人在空间中的y坐标

6.9.3 方法信息

方法名（输入）：返回类型	方法说明
return_base_state():struct	其他类可以返回获得机器人底盘的属性
user_control_base(struct):void	用户用于控制机器人底盘的方法
robot_control_base(struct):void	机器人自主控制底盘的方法

6.10 SlamMap 类

6.10.1 类图



SlamMap 类对 slam 室内建图进行管理。

6.10.2 属性信息

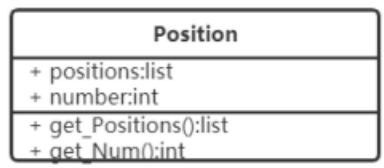
属性名	类型	说明
map	Map	map 为我们利用 slam 算法所得到的当前室内建图，包括已经探明的障碍物、不包含障碍物的区域和还未探索的区域

6.10.3 方法信息

方法名（输入）：返回类型	方法说明
get_Position(): (double, double)	获取当 slam 的建图的坐标
Hector_Slam()	执行 slam 建图操作
update_Map()	更新当前的 map
MapTools()	利用 MapTools 工具来在地图上设置航点位置

6.11 Position类

6.11.1 类图



Position类对物体和目标点的坐标进行管理。

6.11.2 属性信息

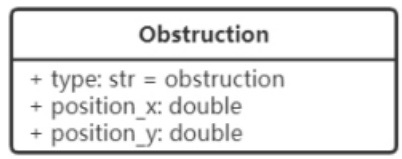
属性名	类型	说明
positions	list	以链表的形式储存障碍物和目标地点的坐标
number	int	不同位置的数量，包括障碍物位置和目标位置

6.11.3 方法信息

方法名（输入，输出）：方法类型	方法说明
get_Positions():list	根据slam类地图的更新结果来获取各位置的坐标
get_Num():int	获取地图中位置的数量，包括障碍物位置和目标位置

6.12 Obstruction 类

6.12.1 类图



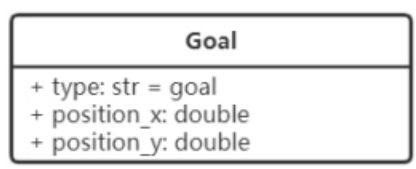
Obstruction 对障碍物的信息进行管理。

6.12.2 属性信息

属性名	类型	说明
type	str=obstruction	type 字段为 obstruction 表示该坐标为障碍物
position_x	double	障碍物的 x 坐标
position_y	double	障碍物的 y 坐标

6.13 Goal 类

6.13.1 类图



Goal 类对目标地点的信息进行管理。

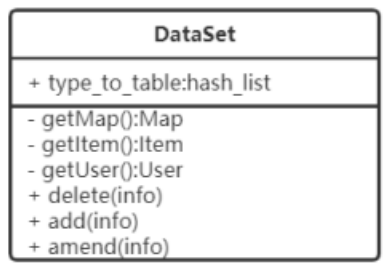
6.13.2 属性信息

属性名	类型	说明
type	str=goal	type 字段为 goal 表示该坐标为目标地点，即机器人应该前往的坐标

position_x	double	目标点的 x 坐标
position_y	double	目标点的 y 坐标

6.14 DataSet 类

6.14.1 类图



Dataset 类对地图、物体和用户信息与数据库的交互进行管理。

6.14.2 属性信息

属性名	类型	说明
type_to_table	hash list	dataset 类主要用来从我们的 SQL 数据库中获取不同类型的数据，因此我们决定用一个 hash 表将不同的请求与 SQL 中不同的 table 对应起来，从而快速找到数据。

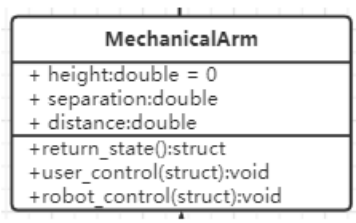
6.14.3 方法信息

方法名（输入，输出）：方法类型	方法说明
getMap():Map	该方法用来获取 Map 类数据用于导航
getItem():Item	该方法用来获取 Item 类数据用于目标物的识别和抓取
getUser():User	该方法用于获取用户类的数据用于登录和区分用户等级

delete(info)	该方法用于删除数据库中的 info 信息
add(info)	该方法用于增加 info 信息到数据库
amend(info)	该方法用于修改数据库中的 info 信息

6.15 MechanicalArm类

6.15.1 类图



MechanicalArm类对于机械臂运动进行控制，其他控制类可以调用其中的控制方法。

6.15.2 属性信息

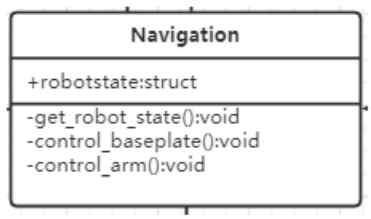
属性名	类型	说明
height	double	机器臂高度
separation	double	机器臂两臂之间宽度
distance	double	机器臂与机器人距离

6.15.3 方法信息

方法名（输入，输出）：方法类型	方法说明
return_state():struct	其他类可以返回获得机器臂的属性
user_control(struct):void	用户用于控制机械臂的方法
robot_control(struct):void	机器人自主控制机械臂的方法

6.16 Navigation类

6.16.1 类图



Navigation类为导航控制类，依赖于数据库和目标识别，在机器人主动控制时，对于底盘和机械臂进行控制导航。

6.16.2 属性信息

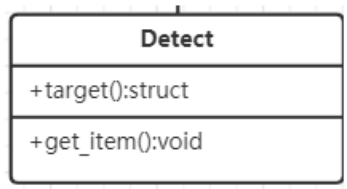
属性名	类型	说明
robotstate	struct	机器人当前状态（包括，底盘和机械臂）

6.16.3 方法信息

方法名（输入，输出）：方法类型	方法说明
get_robot_state():void	获得机器人当前状态属性
control_baseplate(struct):void	用于控制底盘的方法
control_arm(struct):void	用于控制机械臂的方法

6.17 Detect 类

6.17.1 类图



Detect 类为目标识别类，其根据用户的语音或者键盘输入，进行目标物的识别，被 Navigation 类所调用生成的结果。

6.17.2 属性信息

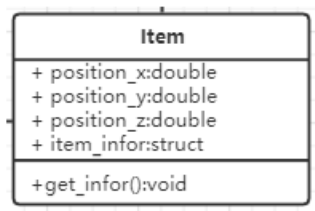
属性名	类型	说明
target	struct	抓取物品的目标信息

6.17.3 方法信息

方法名（输入，输出）：方法类型	方法说明
get_item():void	从用户控制类中读取物品名称，并识别相应物体，生成目标物的三维坐标等信息，赋给类的 target 属性。

6.18 Item类

6.18.1 类图



Item类包括物体的三维坐标和形状标签等对应信息。

6.18.2 属性信息

属性名	类型	说明
position_x	double	物品的坐标信息
position_x	double	物品的坐标信息
position_x	double	物品的坐标信息
item_infor	struct	物品形状对应标签等信息

6.18.3 方法信息

方法名（输入，输出）：方法类型	方法说明
get_infor():void	从Detect类中获得关于物体的信息

7. 运行与开发环境

7.1 运行环境

7.1.1 硬件环境：

表 7-1 硬件运行环境表

名称	数量	参数/功能
启智 ROS 机器人	1	重量约为 30kg（包含抓取组件），承载能力 10kg
机载平板电脑	1	运行 ROS 操作系统，通过 USB 接口与机器人底盘内的 USB_HUB 连接。
视觉传感器	1	kinect2 视觉传感器
激光雷达	1	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	1	启智控制器内置 MPU6050 惯性测量传感器
启智伺服电机模块	1	内置驱动控制板，电流环伺服周期为 50μs，速度环、位置环伺服周期为 1ms。
启智控制器	1	通过 RS485 总线与启智伺服电机模块通讯
启智电池模块	1	正常工作输出电压范围为 23.1V 至 29.4V
轮子	3	全向轮
机械手臂	1	用于抓取桌面上物品的机械臂，该机械臂提供两个控制量： 上升高度和手爪的闭合宽度。

7.1.2 硬件工作环境：

1. 硬质工作平面，可承载不少于 40kg 的重量，坡度不大于 15 度。
2. 温度 15° C~35° C。
3. 远离雾，地面积水，雨水及任何其他液体。

7.1.3 软件环境：

系统环境：kenetic 版本 ROS 系统，基于 Ubuntu16.04

软件包：

表 7-2 软件运行环境表

软件包	说明	版本限制
启智 ROS 源代码	ROS 基本使用代码	kinetic 版本
OpenCV	计算机视觉库	2.4.X
cmake	安装（编译）工具	无
PCL	点云相关开源 C++编程库	1.7.X
libusb	USB 驱动	>=1.0.20
TurboJPEG	图像编辑解码器	无
OpenGL	开放式图形库，用于渲染 2D、3D 矢量图形	无
OpenCL/CUDA/VAPPI	GPU 加速（不同 GPU 芯片可选不同软件包）	无
OpenNI2	开放自然交互的框架	无
科大讯飞在线语音识别包	语音识别交互	kinetic 版本
导航地图工具包	地图导航	无
RPLIDAR ROS	激光雷达驱动	无

7.2 开发环境

7.2.1 硬件环境：

表 7-3 硬件开发环境表

处理器	i7-4720HQ/i5-7400
内存	8GB
系统类型	64 位操作系统，基于 x64 的处理器
GPU	NVIDIA GTX960M

7.2.2 软件环境：

表 7-4 软件开发环境表

操作系统	Ubuntu 16.04
	ROS Kinetic
IDE	Robo Ware Studio
python 功能支持（调试）	pylint
clang-format 功能支持（自动化整理代码）	clang-format3.8 及以上版本

8. 需求可追踪性说明

8.1 功能需求

此机器人所涉及的硬件主要包括鼠标、键盘、麦克风等用于用户交互的硬件，以及启智机器人所包含的用于其捕捉信息和行进运动的硬件，主要有运动底盘、360° 激光雷达、1080P 高清摄像机、TOF 立体相机和 USB-HUB 所拓展的一些设备，如：音响，Kinect2，思岚激光雷达 A2，WIFI 路由器，面板接口等。这些硬件设备将支持此服务型机器人的运行。

8.1.1 基本建图功能

软件：SlamMap 类

硬件：360° 激光雷达。

8.1.2 机器人主动控制

软件：Control 类，BasePlate 类

硬件：运动底盘。

8.1.3 导航功能

软件：Goal 类，Obstruction 类，Control 类，BasePlate 类，DataSet 类

硬件：运动底盘，360° 激光雷达。

8.1.4 物品识别与抓取功能

软件：MachineArm 类，Item 类，Detect 类

硬件：机械臂，1080P 高清摄像机、TOF 立体相机。

8.1.5 语音交互

软件：Speech 类

硬件：面阵麦克风，立体扬声阵列。

8.2 非功能需求

8.2.1 性能指标

- 响应时间：在 95%的情况下,能在 1s 内对一般指令做出反应，能在 5~10s 内对语音指令做出反应。

  设计考虑：在迭代开发中不断优化算法，提高系统实时响应速率。
- 功耗：机器人电源由电池模块供给,工作电压 24V,额定功率 14W,持续工作电流 1.4A,可以持续工作 4 小时以上。

设计考虑：降低算法复杂度，避免系统长时间高负荷运转。

- 处理能力要求：机器人内部的信息具有较高的更新率和反馈率；能在较短时间内对用户期望的目标进行识别和反应，快速避障；具有较高的精准度,能够准确移动到目标位置,准确识别,抓取物体,误差<=0.1m。

  设计考虑：在迭代开发中不断优化算法，提高建图准确率，物体识别准确率，抓取成功率，语音识别准确率；路径规划时避免出现摆荡，确保机器人平稳移动。

8.2.2 质量属性

- 可用性：控制指令简单，易于用户操作；具有一定容错能力；提供数据备份和恢复功能。

  设计考虑：界面操作详细设计见接口设计中“对外用户调用接口”。
- 安全性：区分用户权限，实现不同的功能；记录运行日志，记录运行中的系统错误和用户关键操作信息。

  设计考虑：区分普通用户和管理员，一些特定权限仅管理员具有。
- 可移植性：仅支持 Ubuntu16.04 及 ROS Kinetic 环境下运行，暂不支持移植。
- 可维护性及可扩展性：保留历史各版本的源代码；系统功能模块化；限定修改和 bug 修复时间。有清晰的系统结构，规范。

  设计考虑：在迭代开发中保存历史版本的文档和代码。