**家政服务机器人小智**

**软件设计说明书**

**SDD501**

**V1.7**

分工说明

|  |  |  |
| --- | --- | --- |
| 小组名称 | 拿得稳放得队 | |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 18373292 | 彭维崑 | 编写文档和绘图 |
| 18373727 | 朱自勤 | 文档审核 |
| 18373109 | 孔祥浩 | 内容讨论 |
| 18373248 | 范迪皓 | 内容讨论，前端设计 |
| 18373674 | 汪鸿昊 | 内容讨论 |

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V0.1 | 2021.4.17 | 彭维崑 | 朱自勤 | 除异常处理外的内容 |
| V1.0 | 2021.4.17 | 彭维崑 | 朱自勤 | 更新启动、导航和架构图 |
| V1.1 | 2021.4.20 | 彭维崑 | 朱自勤 | 更新了架构和异常处理 |
| V1.2 | 2021.4.28 | 彭维崑 | 朱自勤 | 更新了系统结构和系统维护功能 |
| V1.3 | 2021.4.29 | 彭维崑 | 朱自勤 | 更新系统结构和接口定义 |
| V1.4 | 2021.4.29 | 彭维崑 | 朱自勤 | 更新前端接口定义 |
| V1.5 | 2021.5.20 | 彭维崑 | 范迪皓 | 更新服务器端返回异常状态码，更新服务器端异常处理。 |
| V1.6 | 2021.6.9 | 彭维崑 | 朱自勤 | 更新异常处理 |
| V1.7 | 2021.6.18 | 彭维崑 | 朱自勤 | 更新异常处理 |

目录

[1 范围 1](#_Toc74134742)

[1.1 项目概述 1](#_Toc74134743)

[1.2 文档概述 1](#_Toc74134744)

[1.3 术语和缩略词 2](#_Toc74134745)

[1.3.1 术语 2](#_Toc74134746)

[1.3.2 缩略词 2](#_Toc74134747)

[1.4 引用文档 2](#_Toc74134748)

[2. 需求概述 2](#_Toc74134749)

[3. 数据库设计 3](#_Toc74134750)

[3.1 地图信息 3](#_Toc74134751)

[4. 体系结构设计 4](#_Toc74134752)

[4.1 系统层次设计 4](#_Toc74134753)

[4.1.1 用户层 4](#_Toc74134754)

[4.1.2 调度层 4](#_Toc74134755)

[4.1.3 控制层 5](#_Toc74134756)

[4.1.4 功能层 5](#_Toc74134757)

[4.2 系统结构设计 5](#_Toc74134758)

[5. 接口设计 6](#_Toc74134759)

[5.1 用户界面设计 6](#_Toc74134760)

[5.1.1 用户主界面 6](#_Toc74134761)

[5.1.2 地图选择界面 8](#_Toc74134762)

[5.1.3 建图界面 9](#_Toc74134763)

[5.1.4 标注界面 11](#_Toc74134764)

[5.1.5 服务界面 13](#_Toc74134765)

[5.1.6 后端状态码定义 14](#_Toc74134766)

[5.2 调度层接口 15](#_Toc74134767)

[5.3 控制层接口 15](#_Toc74134768)

[5.3.1 控制模块接口 15](#_Toc74134769)

[5.3.2 传感器模块接口 16](#_Toc74134770)

[5.3.3 地图模块接口 17](#_Toc74134771)

[5.4 功能层接口 17](#_Toc74134772)

[5.4.1 雷达单元 17](#_Toc74134773)

[5.4.2 摄像头单元 18](#_Toc74134774)

[5.4.3 麦克风单元 18](#_Toc74134775)

[6. 详细设计 18](#_Toc74134776)

[6.1 地图构建 18](#_Toc74134777)

[6.2 地图标注 20](#_Toc74134778)

[6.3 服务模式 21](#_Toc74134779)

[6.3.1 启动服务 22](#_Toc74134780)

[6.3.2 语音识别 22](#_Toc74134781)

[6.3.3 导航 23](#_Toc74134782)

[6.3.4 物品抓取 24](#_Toc74134783)

[6.3.5 物品释放 25](#_Toc74134784)

[6.3.6 语音回复 26](#_Toc74134785)

[6.4 异常检测与处理 27](#_Toc74134786)

[6.4.1 障碍物检测 27](#_Toc74134787)

[6.4.2 异常姿态检测 28](#_Toc74134788)

[6.4.3 物品识别失败 29](#_Toc74134789)

[6.4.4 后端异常 30](#_Toc74134790)

[6.4.5 命名冲突异常 31](#_Toc74134791)

[6.4.6 命名长度异常 31](#_Toc74134792)

[6.4.7 非法字符异常 32](#_Toc74134793)

[6.4.8 频繁点击按钮异常 32](#_Toc74134794)

[7. 系统维护 33](#_Toc74134795)

[7.1 系统升级 33](#_Toc74134796)

[7.2 恢复出厂设置 33](#_Toc74134797)

[8. 运行与开发环境 34](#_Toc74134798)

[8.1 运行环境 34](#_Toc74134799)

[8.2 软件环境 34](#_Toc74134800)

[9. 需求可追踪性说明 35](#_Toc74134801)

# 1 范围

## 项目概述

随着计算机和人工智能技术的蓬勃发展，人们的生活正在逐步“智能化”。而结合人工智能技术的机器人的广泛应用，正是这种智能化的重要表现之一。除了家中能够自动识别房间地形并完成清洁任务的扫地机器人外，在超市、银行和展馆等领域，智能机器人正越来越多地协助人们完成信息查询和位置导引等任务。

但是，目前市场上现有的智能机器人在普通家庭中的应用仍然十分有限，主要集中在扫地机器人、儿童教育机器人和智能家居机器人等应用方向上。现有的机器人还没有实现智能取物等功能，这与真正应用于家政服务的目标之间还有很长的距离。

基于此，本项目计划开发一种服务机器人“小智”。该服务机器人具有室内等相对固定的房间场景下的手动建图、导航、动态避障、语音指令取物、可定制语音指令以及语音交流等功能。不仅如此该服务机器人具有很强的通用性，只需稍作定制便可用于不同的室内场景。相信该服务机器人的实现能够推动智能机器人在家政领域的应用取得进一步的发展。

## 文档概述

本文档用于：

1. 明确系统的设计与组成部分，为后续开发指明方向。
2. 提高工程可见度，更方便控制管理开发进度。
3. 方便后续测试维护，以及新功能迭代，提供重要的参考信息。

文档分为以下几个部分：

1. 数据库设计
2. 体系结构设计
3. 接口设计
4. 详细设计

## 术语和缩略词

### 术语

本文档涉及的术语有：

* Ubuntu: 一种Linux发行版；
* C++: 一种编程语言；
* Rviz: 一种机器人可视化平台；
* Gazebo: 一种机器人仿真平台。

### 缩略词

本文档涉及的缩略词及全称有：

表格 1 缩略词/全称对应表

|  |  |
| --- | --- |
| 缩略词 | 全称 |
| ROS | Robot Operating System |
| LTS | Long Term Support |

## 引用文档

1. 北京六部工坊科技有限公司，《启智ROS版\_开发手册\_20181109》，2018-11.9
2. 北京六部工坊科技有限公司，《启智ROS版\_实验指导书\_1.3.2》，2020-4.25
3. SRS-软件需求规格书

# 需求概述

表格 2 需求概述表

|  |  |
| --- | --- |
| 需求概述 | |
| 用户需求 | * + - 1. 选择机器人的三种工作模式：建图模式、标注模式、服务模式。       2. 根据实际环境选择地图并做标注。       3. 语音控制机器人进行取物。       4. 与机器人进行语音交流。 |
| 系统需求 | 能够指导机器人完成自动建图、自主导航、实体识别、物品取放、语音识别、显示反馈信息。  能进行系统升级和恢复出厂设置。 |
| 异常处理 | 1. 检测到障碍物，停止并等待，超出固定时限则触发报警，再超出固定时限则结束任务； 2. 识别物品失败，重试一定次数，超过则返回起始坐标，结束本次任务； 3. 语音指令无法识别，提示用户无法识别； 4. 用户界面前端设置了超时熔断机制，避免后端长时间不能响应。 |
| 数据库 | 存储地图信息和用户操作记录，与标注模式和服务模式对接。 |

# 数据库设计

## 地图信息

地图信息主要包括地图名称、地图文件路径、标注文件路径。每个地图都有一个唯一的名称，用String表示。为了服务于不同的地图场景，应对标记地点变化，每一个地图还存储了对应的地图文件路径和对应的标注文件路径，给用户提供地图选择和标注的服务。

表格 3 地图信息数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 能否为空 | 自动递增 | 备注 |
| Map\_name | String | NULL | 否 | 否 | 地图名称 |
| Map\_path | String | 根目录 | 否 | 否 | 地图路径 |
| Index\_path | String | 根目录 | 是 | 否 | 标注路径 |
| Map\_show\_path | String | 根目录 | 否 | 否 | 地图图片路径 |

# 体系结构设计

## 系统层次设计

总体的系统层次自顶向下分为用户层、调度层、控制层、功能实现层和硬件层完成。

电脑萤幕和键盘

描述已自动生成

图 1 系统架构图

### 用户层

在用户层，用户与用户界面进行交互，一方面能够通过标注模式对地图进行操作，同时能够选择进入建图模式和服务模式，向调度层发送启动对应服务的指令，同时，程序会根据数据库存储的地图信息向调度层载入地图。

### 调度层

在调度层，总调度单元会根据用户界面发送过来的服务请求，启动对应的服务，接收传感器模块的反馈信息，并由此向总控制模块发送控制信号。

### 控制层

在控制层，总控制模块主要根据调度单元分配的任务向不同的输出单元发送子任务参数，如向运动控制单元发送机器人移动参数，向机械臂单元发送机械臂运动参数，向扬声器单元发送语音输出内容。传感器模块则负责处理不同输入单元发送的信号，如雷达信号，摄像头信号，麦克风信号等，并向总调度单元发送处理结果如输入的语音内容，物体识别成功与否等消息。

### 功能层

在功能层，不同的功能单元与硬件层紧密结合，负责控制硬件完成最基本的任务，如机器人移动，机械臂移动，语音播报采集传感器信息等。

## 系统结构设计

将系统层次中的主要模块进行封装，得到系统结构图。

图片包含 图形用户界面

描述已自动生成

图 2 系统设计类图

从UML类图的角度，重新刻画了系统的结构，同时给出了各个模块的接口，对于接口的具体设计将在下一节说明。

# 接口设计

## 用户界面设计

### 用户主界面

图形用户界面, 应用程序

描述已自动生成

图 3 用户主界面设计图

用户主界面上主要显示用户可以执行的三个功能，建图、地图标注和请求服务。顶部右上角有一个设置按钮，有系统维护的两个选项，系统升级和恢复出厂设置，如下图所示。



手机屏幕的截图

描述已自动生成



图 4 系统维护功能

系统维护界面能够随时进行恢复出厂设置功能，用户收到系统升级通知后也能点击按钮进行系统升级。

表格 14系统升级接口定义

|  |  |
| --- | --- |
| 系统升级 | |
| 后端接口 | /map/update-service |
| 请求方式 | post |
| 参数 | 无 |
| 操作 | 后端调用脚本，完成更新操作。 |
| 返回 | msg，成功时code为200，失败时返回对应错误码400。 |

表格 15 恢复出厂设置接口定义

|  |  |
| --- | --- |
| 恢复出厂设置 | |
| 后端接口 | /map/remove-all |
| 请求方式 | post |
| 参数 | 无 |
| 操作 | 清空数据库表map |
| 返回 | msg，成功时code为200，失败时返回对应错误码602. |

### 地图选择界面

图形用户界面, 应用程序, Teams

描述已自动生成

图 5 地图选择界面设计图

地图选择界面显示目前存储的所有地图，用户进入标注模式和服务模式时首先要进入地图选择界面选择地图。选择地图后机器人自动进入标注模式或者是服务模式。

表格 4 显示地图接口定义

|  |  |
| --- | --- |
| 显示地图 | |
| 后端接口 | /map/display-all |
| 请求方式 | post |
| 参数 | 无 |
| 操作 | 访问数据库，并将map表中的Map\_name、Map\_ show\_path以元组列表形式发送给前端。 |
| 返回 | msg，成功时code为200，失败时对应错误码为604。 |

### 建图界面

图形用户界面, 应用程序, Teams

描述已自动生成

图 6 建图界面设计

建图界面可以实时显示用户建立的地图，用户点击保存地图会提示用户输入地图名称等相关信息，将地图保存在指定位置。

表格 5 开始建图接口定义

|  |  |
| --- | --- |
| 开始建图 | |
| 后端接口 | /control/create-map |
| 请求方式 | post |
| 参数 | 无 |
| 操作 | 调用建图脚本，启动机器人建图服务。 |
| 返回 | msg，成功时code为200，失败时返回对应错误码400。成功后前端跳转到下一界面。 |

表格 6 保存地图接口定义

|  |  |
| --- | --- |
| 保存地图 | |
| 后端接口 | /control/save-map |
| 请求方式 | post |
| 参数 | 无 |
| 操作 | 调用地图保存脚本，将地图文件保存到ROS默认的路径，然后停止执行建图脚本。 |
| 返回 | msg，成功时code为200，失败时返回对应错误码601。成功后跳转到下一页面。 |

表格 7 重命名地图接口定义

|  |  |
| --- | --- |
| 重命名地图 | |
| 后端接口 | /map/insert |
| 请求方式 | post |
| 参数 | 地图名称Map\_name、地图路径Map\_location、可视地图路径Map\_show\_path。默认文件系统下，ROS地图路径位于后端./map/map\_message路径下，可视地图位于前端./assets/map/map\_view路径下。 |
| 操作 | 操作：访问数据库，如果出现重名地图则反馈给前端错误码603，不存在重名地图则将地图名称和两个对应文件系统路径存入数据库的map表中，标注路径暂时为null。后端将地图文件从默认路径拷贝到保存路径，并调用脚本将地图转换为可视化格式文件存储到可视地图路径。 |
| 返回 | 返回：msg，成功时code为200，失败时返回对应错误码603或630。成功后跳转到下一界面，并在前端传递地图名称。 |

### 标注界面

图形用户界面, 应用程序

描述已自动生成

图 7 标注界面设计

标注界面用户能够标注多个航点，如放水的点，放茶的点等，使得小智能够完成更加复杂的任务。

表格 8 开始标注接口定义

|  |  |
| --- | --- |
| 开始标注 | |
| 后端接口 | /control/mark-index |
| 请求方式 | post |
| 参数 | 地图名称name |
| 操作 | 在数据库查询地图所对应的pgm文件路径，并拷贝到ROS工作的默认地图路径处，调用标注航点脚本，启动RVIZ。 |
| 返回 | 返回：msg，成功时code为200，失败时返回对应错误码604、630或400。 |

表格 9 保存标注接口定义

|  |  |
| --- | --- |
| 保存标注 | |
| 后端接口 | /control/save-index |
| 请求方式 | post |
| 参数 | 无 |
| 操作 | 调用保存航点脚本，保存航点信息，其间需要保持RVIZ开启状态。 |
| 返回 | msg，成功时code为200，失败时返回对应错误码605或640。 |

表格 10 显示航点列表接口定义

|  |  |
| --- | --- |
| 显示航点列表 | |
| 后端接口 | /control/index-count |
| 请求方式 | post |
| 参数 | 无 |
| 操作 | 后端读取ROS工作目录中的默认航点文件，解析出标注的航点名称列表反馈给前端。 |
| 返回 | msg，成功时code为200，失败时返回对应错误码609。data字段为原航点名称的列表。 |

表格 11 重命名航点接口定义

|  |  |
| --- | --- |
| 重命名航点 | |
| 后端接口 | /control/rename-mark |
| 请求方式 | post |
| 参数 | 地图名称Map\_name，航点路径index\_location(默认路径设置为后端./map/map\_index路径下)，原航点名称与新航点名称对应的字典indexes。 |
| 操作 | 后端修改航点文件，并将地图的航点文件保存到航点路径，然后将航点文件拷贝一份至航点路径备份。 |
| 返回 | msg，成功时code为200，失败时返回对应错误码为607。 |

### 服务界面

图片包含 图形用户界面

描述已自动生成

图 8 服务界面设计

用户在服务界面能够标注起始位置，同时能够手动选择开始服务或结束服务。

表格 12 开始服务接口定义

|  |  |
| --- | --- |
| 开始服务 | |
| 后端接口 | /control/robot-service |
| 请求方式 | post |
| 参数 | 地图名称Map\_name。 |
| 操作 | 访问数据库，将地图的pgm文件、航点文件均拷贝到ROS的默认工作路径。然后启动RVIZ，用户标注起始点后机器人开始服务。 |
| 返回 | msg，成功时code为200，失败时code为700、604或609。 |

表格 13 结束服务接口定义

|  |  |
| --- | --- |
| 结束服务 | |
| 后端接口 | /control/stop-service |
| 请求方式 | post |
| 参数 | 无 |
| 操作 | 停止服务脚本 |
| 返回 | msg，成功时code为200，失败时code为700或610。 |

### 后端状态码定义

通过定义后端的状态码，给前端准确的信息反馈。

表格 15 后端状态码定义表

|  |  |
| --- | --- |
| 200 | 成功 |
| 400 | 一般异常 |
| 601 | 保存地图异常 |
| 602 | 删除地图异常 |
| 603 | 地图名称冲突异常 |
| 604 | 地图不存在异常 |
| 605 | 保存航点异常 |
| 606 | 删除航点异常 |
| 607 | 修改航点异常 |
| 608 | 航点名称冲突异常 |
| 609 | 航点不存在异常 |
| 610 | 脚本执行通用异常 |
| 611 | 路由空指针属性异常。 |
| 612 | Linux命令进程冲突异常。 |
| 630 | 地图文件操作异常 |
| 640 | 航点文件操作异常 |
| 700 | 机器人服务通用异常 |

## 调度层接口

调度层负责向用户界面提供执行用户请求的接口，与提供给用户的服务功能一一对应，即建图、取物和语音交流三项服务。具体来说，就是根据用户的输入启动不同的任务脚本，分别对应建图、取物和语音交流三种功能，根据任务脚本的逻辑调用控制器和传感器执行不同的子任务。

表格 16 调度层接口定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 订阅 | 发布 | 说明 |
| slam\_service | 无 | 无 | 手动建图任务总体逻辑 |
| main\_service | 无 | 无 | 取物服务和语音交流总体逻辑 |
| mark\_service | 无 | 无 | 标注服务总体逻辑 |

## 控制层接口

控制层分为总控制模块和传感器模块。根据调度层的服务进行拆分并结合下层通信的硬件，这两个模块将服务拆分为不同的子任务，并将这些子任务交由不同的功能模块完成。

### 控制模块接口

控制模块的接口对应了包括了导航、物品抓取和释放、语音回复。这些接口将对应的任务实现逻辑进行封装，并根据实现逻辑发送控制消息给功能单元协调完成。

表格 17 控制模块接口定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 订阅 | 发布 | 说明 |
| navigation | /map  /goal  /LaserScan | /cmd\_vel | 封装了导航逻辑，即move\_base节点 |
| reply | /SoundRequest | /robotsound | 封装了语音输出逻辑，即sound\_play |
| grab | /pointcloud | /wpb\_home\_mani\_ctrl | 封装了取物逻辑，即wpb\_home\_grab\_server节点 |
| pass | 无 | /wpb\_home\_mani\_ctrl | 封装了放物逻辑，即wpb\_home\_pass\_server节点 |

### 传感器模块接口

传感器模块的接口对应了环境扫描、语音识别、姿态感知、。这些接口将对应的任务实现逻辑进行封装，并订阅了不同的传感器话题，将这些传感器发布的消息分别进行处理，根据这些功能的逻辑将消息发布给控制模块的节点或调度模块的节点。

表格 18 传感器模块接口定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 订阅 | 发布 | 说明 |
| vicinity\_rec | /LaserScan | /nav\_msgs/OccupancyGrid | 封装了建图总体逻辑，即gmapping节点 |
| pose\_check | /imu | /odom\_combined | 封装了检测机器人姿态的逻辑，robot\_pose\_ekf节点 |
| speech\_rec | 用户输入 | /xfyun/iat | 封装了语音识别逻辑，即iat\_node节点 |
| joy\_ctrl | /Joy | /cmd\_vel | 封装了手柄操纵消息转为速度控制消息的逻辑，即joy\_node节点 |

### 地图模块接口

地图模块单列出来，专门提供显示地图和保存地图两种常用的服务。

表格 19 地图模块接口定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 订阅 | 发布 | 说明 |
| open | 无 | 无 | 打开地图，启动map\_server和wp\_manager节点 |
| save | 无 | 无 | 保存地图，随后关闭map\_server和wp\_manager节点 |

## 功能层接口

功能层接口与硬件结合紧密，控制模块下的功能单元完成控制模块下达的指令，进行移动、机械臂控制、语音播报等功能，由于大部分节点都能够直接发布具体的控制指令，不需要功能单元在单独发送，因此控制模块下的功能单元不做接口设计。传感器模块下的功能单元主要负责收集传感器信息并发送给传感器模块集中处理。

### 雷达单元

雷达单元已经集成在了rplidar这个包中。雷达将360°扫描机器人四周，并将扫描的帧等参数发布到sensor\_msgs/LaserScan话题上。

表格 20 雷达单元接口定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 订阅 | 发布 | 说明 |
| scan | 无 | /LaserScan | 封装了激光雷达扫描并输出扫描消息的逻辑，即rpLidarNode和wpb\_home\_lidar\_filter两个节点 |

### 摄像头单元

摄像头使用kinect2，摄像头的参数设置能够在Joint\_State\_Publisher下看到。而kinect2采集的信息包括rgb图和点云图将发布在/kinect2下的主题中。

表格 21 摄像头接口定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 订阅 | 发布 | 说明 |
| capture | 无 | /kinect2 | 封装了摄像头信号相关的处理节点kinect2\_bridge等 |

### 麦克风单元

麦克风单元通过系统调用获得音频信号的输入，这一功能已经集成在语音识别中，这里不再详细说明。

# 详细设计

在前文架构总体设计的基础上，我们组借助已经写好的各种信息处理节点和控制节点对各个任务进行了更细致的设计。在各层通信上并没有采用层次化的通信方式，而是根据ROS独有的通信机制，最终在具有层次的架构设计上实现了更灵活的功能实现。

## 地图构建

地图构建接口主要基于雷达单元，手柄和运动单元完成。用户在用户界面选择建图模式后，用户界面将通知总调度器启动建图服务，即slam\_service()。总调度器启动这项服务后，会首先启动机器人的模型仿真环境程序以及内核wpb\_home\_core，然后调度传感器模块启动激光雷达节点rplidarNode和手柄节点joy\_node。雷达节点将开始扫描周边环境，并通过wpb\_home\_lidar\_filter对原始数据进行滤波处理，将处理的数据发布到/LaserScan话题；同时，joy\_node根据手柄的控制将传感信息发布到sensor\_msgs/Joy话题。传感器调用vecinity\_recog()函数订阅/scan话题，利用gmapping节点构建地图，发布地图数据到/nav\_msgs/OccupancyGrid话题，由调度层订阅话题并实时显示地图；传感器模块同时调用joy\_ctrl()函数订阅sensor\_msgs/Joy话题，利用wpb\_home\_js\_vel节点将手柄传感器数据转化为控制机器人运动的速度参数，通过/cmd\_vel话题传递给运动单元，完成运动的控制。直到用户在用户界面上选择保存地图，将通知总调度器调用地图的open()函数，启动map\_server节点保存地图，同时将地图拷贝一份存入指定的文件夹中，按照用户输入的地图名称命名，并将文件路径等信息写入到数据库中，再向用户反馈地图保存成功这一信息，最后停止整个任务。

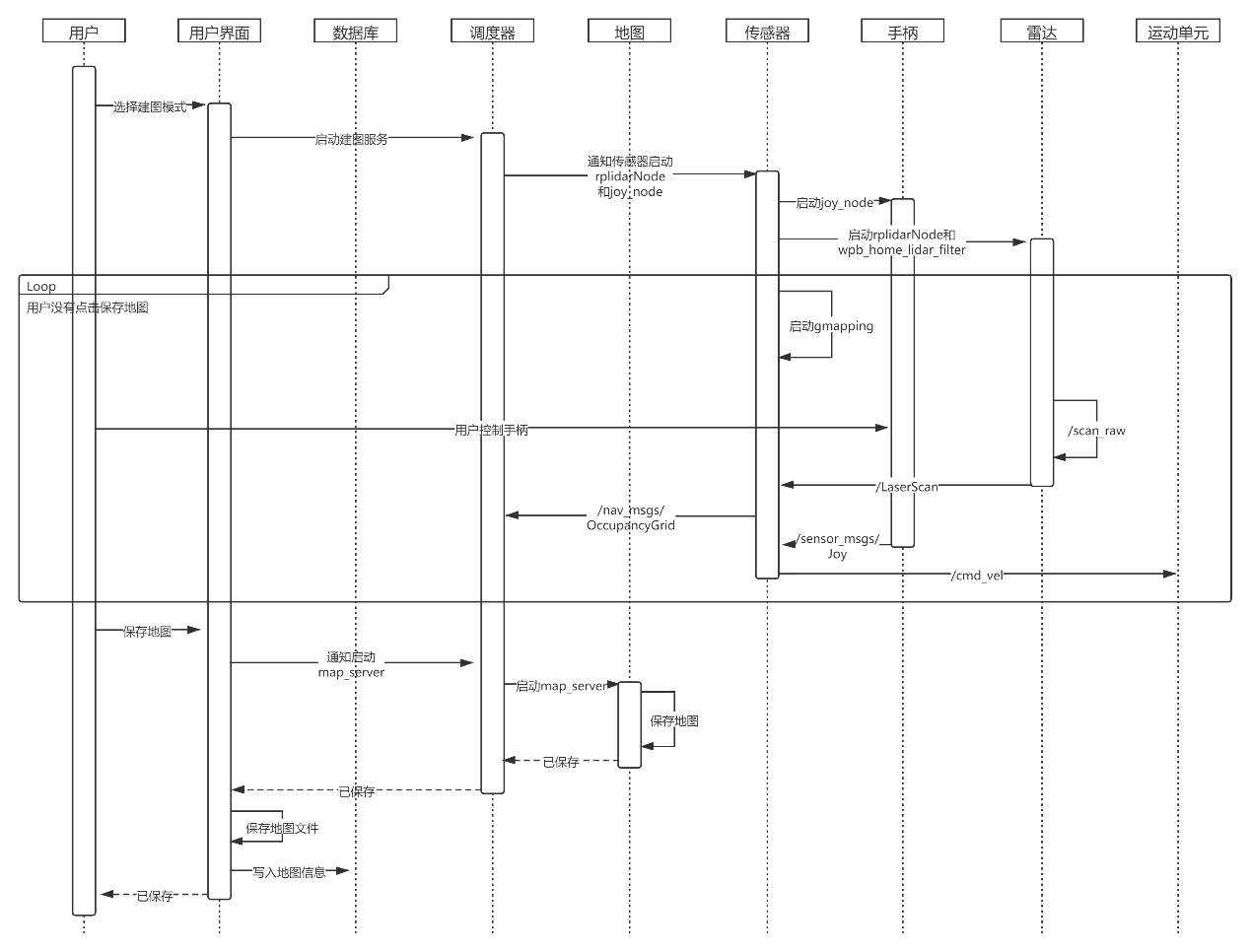


图 9 手动建图时序图

## 地图标注

地图标注接口主要由MapTools工具实现。MapTools工具能够将地图在Rviz下打开，能够直接在地图上标记导航点并保存导航点的数据。因此我们采用MapTools工具实现地图标注的功能。

用户在用户界面选择标注模式，进入到地图选择界面，选择好需要标注导航点的地图后用户界面会根据数据库中存放的对应地图信息找到地图，将地图拷贝到maps文件夹下，然后通知总调度器启动标注服务，即mark\_service()。接着总调度器通知地图调用open()启动map\_server节点和wp\_manager节点，选择的地图将会被直接呈现在用户界面上，用户标注导航点后，在界面上点击保存，将通知总调度器调用地图的save()函数，save()函数运行wp\_saver节点保存导航点数据，同时用户界面将导航点文件拷贝到指定位置，并将文件信息写入数据库，然后通知用户标注完成，结束任务。

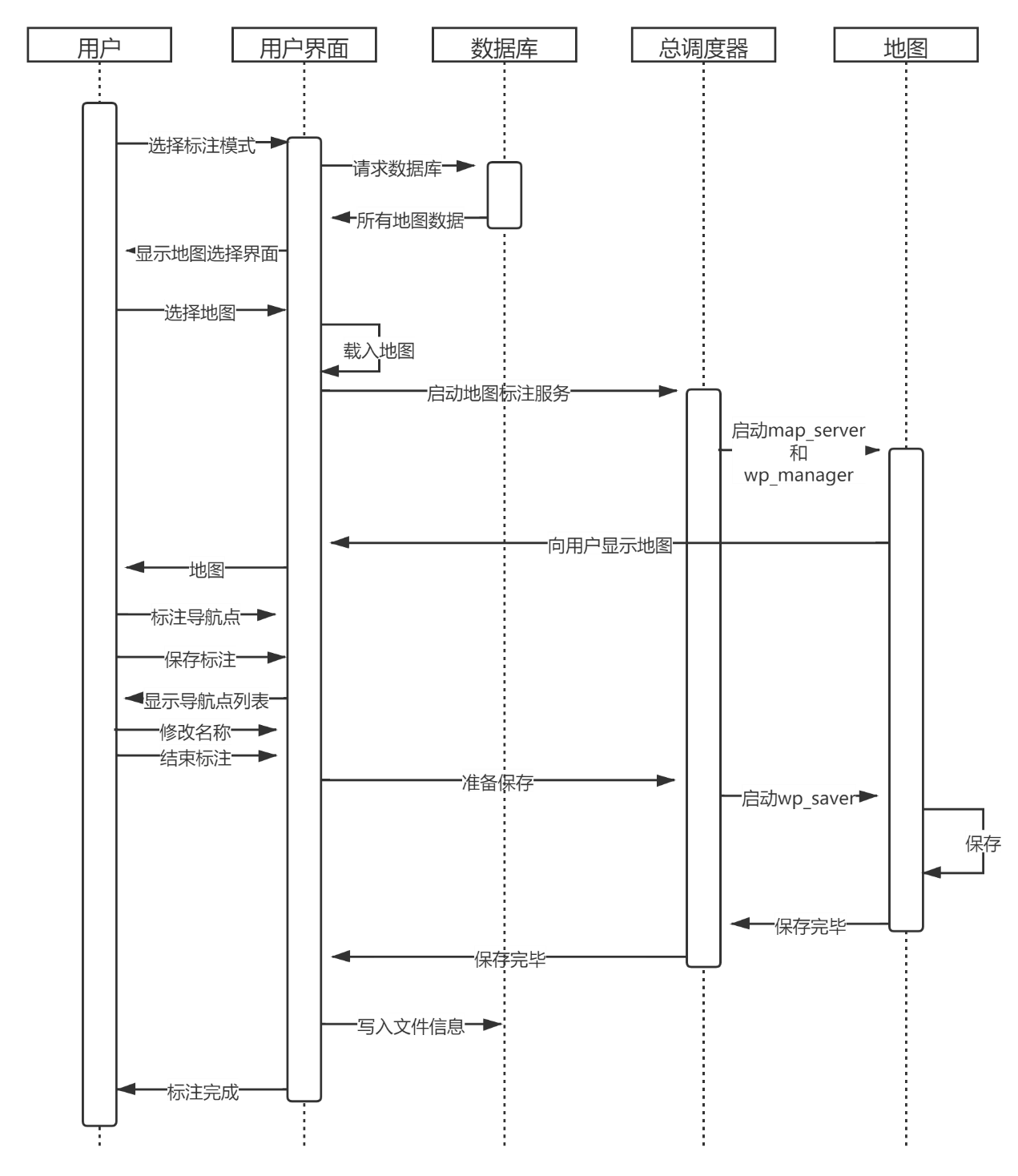


图 10 地图标注时序图

## 服务模式

服务模式内容较多且比较复杂，主要完成物品抓取和语音交流两个主要任务，这两个任务又可以拆分成多个可以复用的子模块，分别为启动服务、语音识别、导航、物品抓取、物品释放、语音回复六个部分。

### 启动服务

用户首先需要启动服务。在主界面选择服务模式后，进入地图选择界面，选好地图后，用户界面将根据选定地图的文件路径将文件载入指定文件夹，然后启动调度器，运行main\_service()函数，启动任务管理系统action\_manager，并调用地图类open()函数启动map\_server节点和wp\_manager节点，向用户展示地图。接下来调用传感器的speech\_recog()函数启动iat\_node，做好随时接受语音指令的准备。在此期间，用户完成对机器人初始位置进行标定，协助机器人确定位置。

截图里有图片

描述已自动生成

图 11 启动服务时序图

### 语音识别

语音识别主要由麦克风单元完成，而麦克风单元的功能已经被集成在了语音识别的节点中，因此不再单独列出。

用户在进入物品抓取服务后，总调度器会首先通知传感器启动语音识别节点iat\_node，做好随时接受语音指令的准备。接受到用户的语音输入后，传感器将识别的语音内容发布到/xfyun/iat话题上，总调度器根据识别内容提取关键词，启动对应的服务。

图示

描述已自动生成

图 12 语音识别时序图

### 导航

导航部分，机器人需要协调地图、雷达单元、运动单元完成。总调度器在获得物品信息后，将物品信息与地图标注比对得到物品坐标。总调度器再通知传感器调用雷达模块的scan()函数，启动rplidarNode和wpb\_home\_lidar\_filter节点，将雷达处理后的数据发布在/sensor\_msgs/LaserScan话题上。同时，总调度器调用控制模块的navigation()函数，启动move\_base节点和amcl节点。地图类将地图信息发布在/map和/nav\_msgs/GetMap话题上，而move\_base节点则订阅雷达和地图发布的话题消息，估算自己的位置并规划路线，将最后计算出的运动参数发布到cmd\_vel话题上，控制运动单元运动;同时还会启动一个MoveBaseClient用于检测导航的状态，如果成功，则向总调度器返回成功，进入下一步。

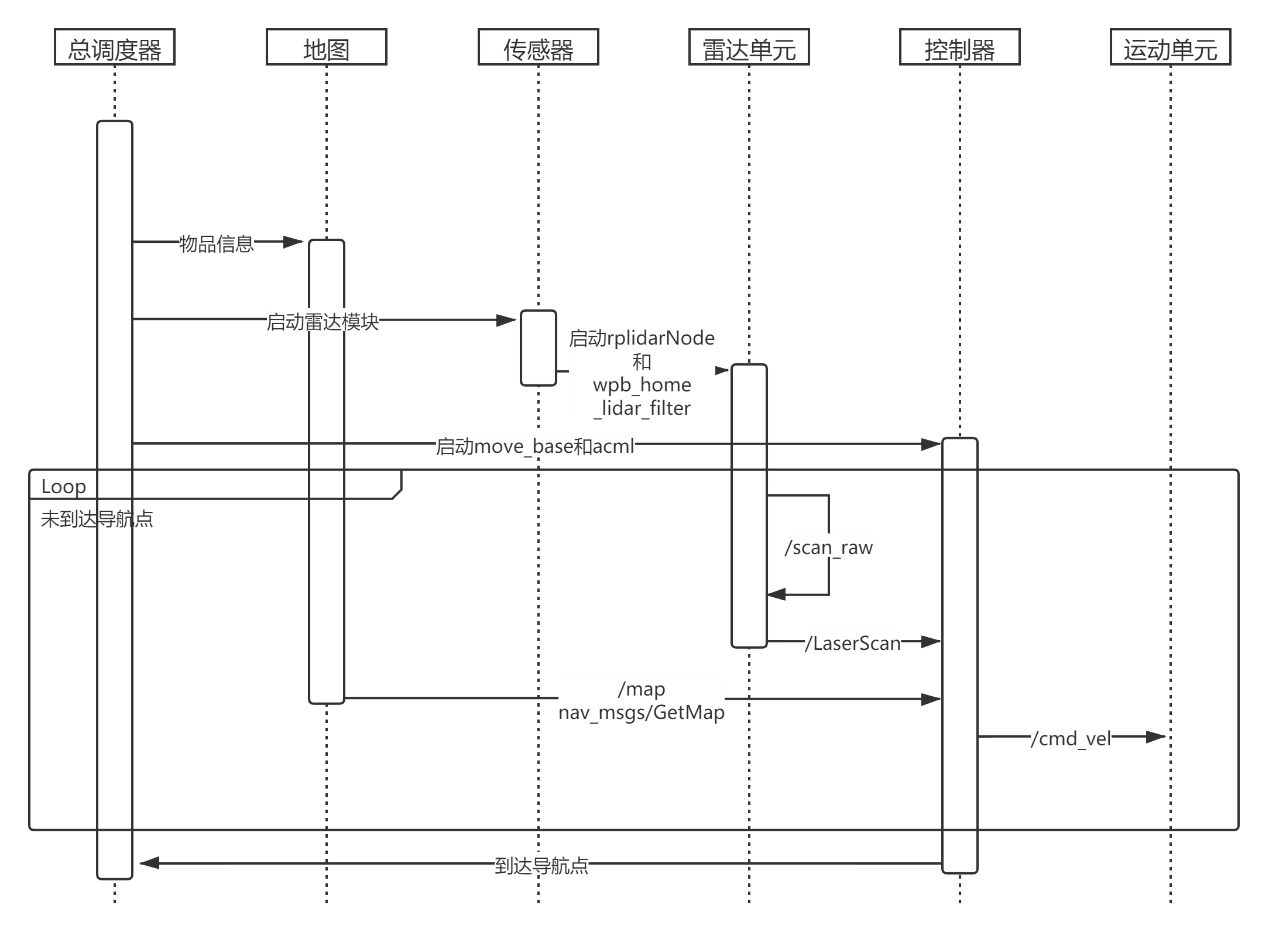


图 13 导航时序图

### 物品抓取

物品抓取是一个较为复杂的任务，需要物品识别和机械臂控制两个任务模块紧密协作完成。具体来说，首先需要摄像头检测桌面，然后配合运动模块调整到桌面的距离，接着检测物品，根据物品位置左右调整位置对准物品，最后操控机械臂抓取物品。

首先，总调度器在检测到导航结束后，调用控制器的grab()函数启动wpb\_home\_grab\_server节点开始取物任务，同时自己启动wpb\_home\_grab\_client节点接收取物的反馈结果。传感器检测到桌面后，控制器就开始发布/cmd\_vel话题调整机器人的位置。调度器再通知传感器进行物品识别，收到识别成功的消息后，控制器发布/cmd\_vel话题调整位置对准物品，然后发布/wpb\_home/mani\_ctrl话题控制机械臂抓取物品。抓取成功后，向后退，并向调度器发布抓取成功的反馈，调度器进入下一阶段，导航到用户坐标。

截图里有图片

描述已自动生成

图 14 物品抓取时序图

### 物品释放

类似的，物品释放就更为简单，不需要摄像头的参与，只需要总调度器调用控制器pass()函数，启动wpb\_home\_pass\_server节点，发布机械臂控制消息/wpb\_home/mani\_ctrl给机械臂，释放物品回复原位即可。调度器会同时启动wpb\_home\_pass\_client节点检测物品释放的反馈。

日程表

描述已自动生成

图 15 物品释放时序图

### 语音回复

语音回复主要由扬声器完成。总调度器生成语音回复内容，调用控制器的reply()函数，启动sound\_play节点，发布/robotsound话题将需要播报的语音内容发送给扬声器进行播报，完成语音播报任务。

图示

描述已自动生成

图 16 语音回复时序图

## 异常检测与处理

机器人在实际运行中会遇到各种各样的问题，不可能像前文规划好的时序流程一步一步完成，为了应对可能发生的异常情况，在需求文档中定义了四种可能的异常情况：机器人检测到障碍物、机器人倾倒、物品检测失败、语音识别失败。对于语音识别部分，用户的非法语音输入在检测到之后机器人会回复特定的语音，与语音回复的实现完全一致，因此不在这里详细描述。接下来主要介绍前三种异常处理的具体流程和实现。

### 障碍物检测

障碍检测的任务主要由雷达完成，这一部分的任务会伴随着导航的进程一起运行，直到导航结束。总调度器在开始执行导航任务的同时，会通知传感器启动障碍物检测节点wpb\_home\_lidar\_obstacle，订阅雷达节点话题/LaserScan并进行检测，一旦发现障碍物将直接发出运动控制话题/cmd\_vel，保证机器人不会撞到障碍物。

但是，如果控制器检测到超时，说明此时障碍物无法避开，则机器人会停止导航并在一段时间内持续发出警报。

日程表

描述已自动生成

图 17 障碍物检测时序图

### 异常姿态检测

姿态检测也是伴随导航任务的处理程序，在机器人导航移动过程中会随时监测机器人的姿态是否正常，如果发现出现了异常的姿态，会在一段时间内持续发出警报。在机器人启动导航的同时，总调度器会通知传感器启动姿态检测节点robot\_pose\_ekf，订阅IMU发出的姿态信息/sensor\_msgs/imu，计算姿态，发现异常时会立即通知总调度器，总调度器会启动控制器的报警，控制器启动sound\_play节点，在/robotsound话题发布报警信息。

图示

描述已自动生成

图 18 姿态检测时序图

### 物品识别失败

物品识别失败出现在物品抓取阶段，可能指定地点没有目标物品或识别出现了错误，需要立即终止物品抓取任务，进入下一阶段任务，即返回用户在坐标，这一处理程序已经集成在wpb\_home\_grab\_server节点中了。在物品识别失败后不断重试，达到一定重试次数后将任务状态设置为完毕，进入下一阶段。

图示

描述已自动生成

图 19 物品识别失败时序图

### 后端异常

用户界面后端出现异常无法响应前端的请求时，前端需要设置一个时限，当超过这一时限就向用户反馈，防止出现长时间等待无响应。

图示

描述已自动生成

图 20 后端异常处理时序图

### 命名冲突异常

用户在给航点或地图命名时可能会出现重复，导致命名冲突，需要提示用户命名冲突，并修改名称。

图示

描述已自动生成

图 21 命名冲突异常处理时序图

### 命名长度异常

用户在命名时可能出现名称过长或为空的情况，需要提示用户修改命名。

图示

描述已自动生成

图 22 命名长度异常处理时序图

### 非法字符异常

用户在命名时可能出现非法字符，需要提示用户修改命名。

图片包含 日程表

描述已自动生成

图 23 非法字符异常处理时序图

### 频繁点击按钮异常

用户在使用过程中可能会出现频繁点击界面某一按钮的情况，需要限制为只响应第一次请求，避免后端由于请求过多而崩溃。

图示

描述已自动生成

图 24 频繁点击异常处理时序图

# 系统维护

在用户需要重置系统，如环境变化，或者有新的功能发布需要进行系统更新时，用户能够通过主界面的系统设置按钮进入系统维护界面，实现这些需求。

## 系统升级

用户点击系统升级按钮后，用户界面会从远端仓库中拉取最新的代码，然后catkin\_make编译完成系统升级。

## 恢复出厂设置

在用户选择恢复出厂设置时，用户界面会删除数据库中所有存储的地图信息和用户操作信息，并删除存储在指定目录下的地图文件和操作记录文件。

# 运行与开发环境

## 运行环境

本系统运行的硬件环境为:

* 启智ROS机器人；
* 启智ROS机器人配套机械臂；
* 互联网连接。

本系统运行的软件环境为：

* Ubuntu 16.04 LTS；
* ROS Kinetic Kame 1.12.17；
* 启智ROS机器人相关基础包和驱动。
* Gazebo仿真环境

在开发阶段，系统主要在仿真环境下运行；在完成每一次迭代后，会将系统在真实的机器人环境下进行运行和测试。

## 软件环境

本系统开发所需的硬件环境为：

* PC开发设备
* 启智ROS机器人；
* 启智ROS机器人配套机械臂；
* 互联网连接。

本系统开发所需的软件环境为

* Ubuntu 16.04 LTS；
* ROS Kinetic Kame 1.12.17；
* 启智ROS机器人相关基础包和驱动；
* Rviz可视化平台与gazebo仿真环境；
* RoboWare Studio集成开发环境。

# 需求可追踪性说明

表格 22 需求追踪表

|  |  |
| --- | --- |
| SRS V2.4文档需求 | SDD V1.6设计内容 |
| 手动建图 | 6.1地图构建 |
| 地图标注 | 6.2地图标注 |
| 导航 | 6.3.3导航 |
| 取物放物 | 6.3.4物品抓取，6.3.5物品释放 |
| 物品识别 | 6.3.4物品抓取 |
| 语音交流 | 6.3.2语音识别，6.3.6语音回复 |
| 异常处理 | 6.4异常检测与处理 |
| 选择界面 | 5.1用户界面设计 |
| 建图界面 | 5.1用户界面设计 |
| 标注界面 | 5.1用户界面设计 |
| 服务界面 | 5.1用户界面设计 |
| 维护界面 | 5.1用户界面设计，7系统维护 |
| 响应时间 | 由硬件决定 |
| 功耗 | 由硬件决定 |
| 处理能力 | 6.4异常检测与处理 |
| 可用性 | 6详细设计 |
| 可移植性 | 4.1系统层次结构设计，7系统维护 |
| 完整性 | 6详细设计 |
| 效率 | 4.1系统层次结构设计，6详细设计 |
| 健壮性 | 6.4异常检测与处理 |