**智能家庭助手**

**需求规格说明书**

**SRS08**

**V1.0**

分工说明

|  |  |  |
| --- | --- | --- |
| 小组名称 | 啊对对队 | |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 18375299 | 范竞元 | 主编 |
| 18375182 | 廖纪童 | 内容审核与修改，内容讨论 |
| 18375200 | 刘裕炜 | 内容讨论 |
| 19373106 | 裴宝琦 | 内容讨论 |
| 18374457 | 刘传 | 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 |
| 1.0 | 2022.4.7 | 范竞元 | 廖纪童 | 初步完成需求分析 |
| 1.1 | 2022.4.13 | 范竞元 | 刘传 | 细化用户界面描述 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[1. 范围 1](#_Toc100265111)

[1.1 项目概述 1](#_Toc100265112)

[1.1.1 系统开发背景 1](#_Toc100265113)

[1.1.2 主要功能 1](#_Toc100265114)

[1.1.3 非功能性需求 1](#_Toc100265115)

[1.1.4 应用场景 2](#_Toc100265116)

[1.2 文档概述 2](#_Toc100265117)

[1.2.1 文档用途 2](#_Toc100265118)

[1.2.2 内容组织 2](#_Toc100265119)

[1.3 术语和缩略词 3](#_Toc100265120)

[1.3.1 术语 3](#_Toc100265121)

[1.3.2 缩略词 3](#_Toc100265122)

[1.4 引用文档 3](#_Toc100265123)

[2. 业务需求 3](#_Toc100265124)

[2.1 用例一：系统功能设置 4](#_Toc100265125)

[2.2 用例一：手动建图 5](#_Toc100265126)

[2.3 用例二：物品抓取 6](#_Toc100265127)

[3. 数据需求 7](#_Toc100265128)

[3.1 数据实体及其关系 7](#_Toc100265129)

[3.2 类建模 8](#_Toc100265130)

[4. 功能需求 9](#_Toc100265131)

[4.1 功能需求概述 9](#_Toc100265132)

[4.2 物品抓取 9](#_Toc100265133)

[4.2.1 导航 9](#_Toc100265134)

[4.2.2 取物/放物 9](#_Toc100265135)

[4.2.3 物品识别 9](#_Toc100265136)

[4.2.4 自动避障 9](#_Toc100265137)

[4.3 语音交流 10](#_Toc100265138)

[4.4 用户界面 10](#_Toc100265139)

[4.4.1 启动界面 10](#_Toc100265140)

[4.4.2 选择界面 10](#_Toc100265141)

[4.4.3 建图界面 10](#_Toc100265142)

[4.4.4 标注界面 10](#_Toc100265143)

[4.4.5 地图界面 10](#_Toc100265144)

[4.4.6 维护界面 10](#_Toc100265145)

[4.5 异常处理 11](#_Toc100265146)

[4.5.1 路径规划失败 11](#_Toc100265147)

[4.5.2 物品识别失败 11](#_Toc100265148)

[4.5.3 语音识别失败 11](#_Toc100265149)

[4.5.4 用户界面异常 11](#_Toc100265150)

[4.6 维护功能 11](#_Toc100265151)

[4.6.1 恢复出厂设置 11](#_Toc100265152)

[4.6.2 移动到新的环境 12](#_Toc100265153)

[5. 非功能需求 12](#_Toc100265154)

[5.1 性能指标 12](#_Toc100265155)

[5.1.1 响应时间 12](#_Toc100265156)

[5.1.2 功耗 12](#_Toc100265157)

[5.1.3 处理能力 12](#_Toc100265158)

[5.2 质量指标 13](#_Toc100265159)

[5.2.1 系统可用性 13](#_Toc100265160)

[5.2.2 可移植性 13](#_Toc100265161)

[5.2.3 完整性 13](#_Toc100265162)

[5.2.4 效率 13](#_Toc100265163)

[5.2.5 健壮性 13](#_Toc100265164)

[6. 运行与开发环境 14](#_Toc100265165)

[6.1 运行环境 14](#_Toc100265166)

[6.2 开发环境 14](#_Toc100265167)

# 范围

## 项目概述

### 系统开发背景

随着物联网、机器人技术的日渐发展，越来越多场景中的设备可以被网络赋予“智能”，从而更方便地协助人。在需求类型较为固定的领域，如银行大厅导览、餐厅送菜等场景，时常能见到工作中的智能机器人。具体到家居方面，目前较为成熟的也有扫地机器人之类功能较为简单的智能家居设备投入应用。

但是，市面上售卖的智能家居大多只局限于联网控制的空调、冰箱、电灯等电器，机器人类也仅限于扫地机器人、教育机器人等。我们希望开发一种家庭服务机器人，能够在房间内根据用户的语音指令帮助用户拿取并运送物品。该机器人可以由用户手动建图后，根据用户的语音指令，自动规划路线取物并送回给用户。该机器人可以让用户不被取物打断手头工作，方便用户的日常生活。

### 主要功能

该服务机器人的主要功能如下：

* 能够手动建立使用场景的室内布局图并保存；
* 能够接受语音指令；
* 能够自动规划路径并导航；
* 能够躲避路径上的障碍物，顺利到达目的地；
* 能够实现物品抓取功能；
* 具备基本的异常处理功能。

### 非功能性需求

该服务机器人希望达到的非功能需求如下：

* 响应速度快；
* 功耗低，可以长时间续航；
* 处理能力强；
* 高系统可用性；
* 高可移植性；
* 功能具备完整性；
* 高运行效率；
* 强健壮性。

### 应用场景

该机器人可以在用户腾不出手或无法抽身时通过语音控制，帮助用户拿取物品。可以应用于日常家居、办公场所等地。

## 文档概述

### 文档用途

本需求规格说明文档用途如下：

* 确定用户对该产品的主要业务需求；
* 确定数据需求，指导下一步开发架构；
* 确定功能需求，指导主要开发方向；
* 确定非功能需求，指导次要开发方向；
* 确定用户界面需求，指导人机交互方向开发工作；
* 为未来测试验收提供指导思路。

### 内容组织

本文档主要包含了以下几部分内容：

* 业务需求
* 数据需求
* 功能需求
* 非功能需求
* 用户界面需求

## 术语和缩略词

### 术语

本文档所涉及到的术语有：

* Ubuntu 18.04 LTS： Linux操作系统的其中一个图形化界面发行版，版本号18.04；
* 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

本文档所涉及到的缩略词及其全称有：

**表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. SDP-软件开发计划

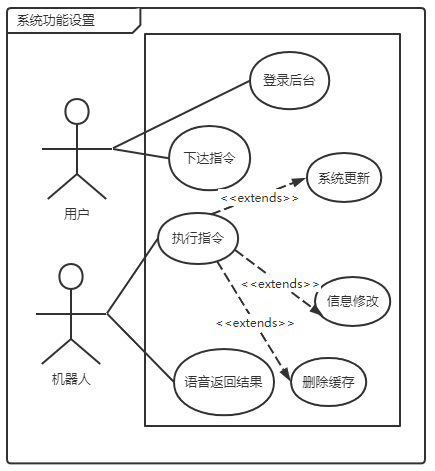
# 业务需求

作为一款家庭服务机器人，我们致力于为客户提供人性化、多功能的服务，目前我们提出了系统功能设置、手动建图、语音识别、图像识别、路径规划、物品取放、人机交互等多个应用场景，在下文我们将会针对系统功能设置、手动建图、物品取放三个场景用例进行分析。

## 用例一：系统功能设置

**表4 系统功能设置用例表**

|  |  |
| --- | --- |
| 主要参与者 | 机器人 |
| 前置条件 | 用户已经启动机器人并且机器人属于正常待机状态 |
| 启动 | 用户与机器人交流 |
| 场景 | 1. 用户开启机器人； 2. 用户界面显示进入“用户模式”或者“后台模式”； 3. 用户选择进入“后台模式”； 4. 用户登录账户； 5. 用户发出后台更新、删除缓存地图、修改账户信息等命令； 6. 机器人执行命令并语音返回结果； |
| 异常分析 | 1. 系统修改信息异常 2. 用户不能正确获得权限以及执行指令 |
| 正常效果 | 用户正常进入后台并进行修改 |
| 何时可用 | 第一次迭代 |
| 使用频率 | 低 |
| 次要参与者 | 用户 |

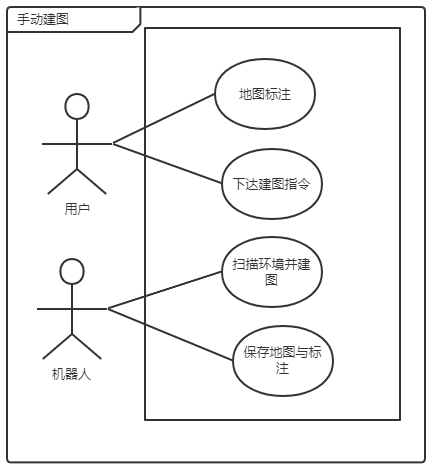
******

**图4 系统功能设置用例图**

## 用例一：手动建图

**表2 手动建图\_用例表**

|  |  |
| --- | --- |
| 主要参与者 | 机器人 |
| 前置条件 | 用户启动机器人且机器人处于有电状态 |
| 启动 | 用户要求机器人对房间建图 |
| 场景 | * + - 1. 用户开启机器人并登录；       2. 用户选择进入“建图模式”；       3. 用户界面显示“建图进行中”，机器人通过自带的传感器扫描周围并记录地形信息；       4. 用户对是否建图完成进行判断；       5. 如果没有完成建图，则用户将机器人移动到下一地点继续建图；       6. 如果完成建图，则机器人将地形图保存下来，并停止移动和建图；       7. 完成建图后，用户界面显示“建图完成”，用户可以为地图命名并标注物品和自己所在的大致位置，或者用户可以之后再在“标注模式”中对地图进行标注；       8. 用户界面退回到开始的选择界面。 |
| 异常情况分析 | * + - 1. 建图过程中机器人被障碍物阻挡而倾倒；       2. 机器人建图不正确。       3. 用户标注地图失败。 |
| 正常效果 | 1.机器人成功对环境建图。  2.用户可以对地图进行正常标注。 |
| 何时可用 | 第一次迭代 |
| 使用频率 | 中 |
| 次要参与者 | 用户 |

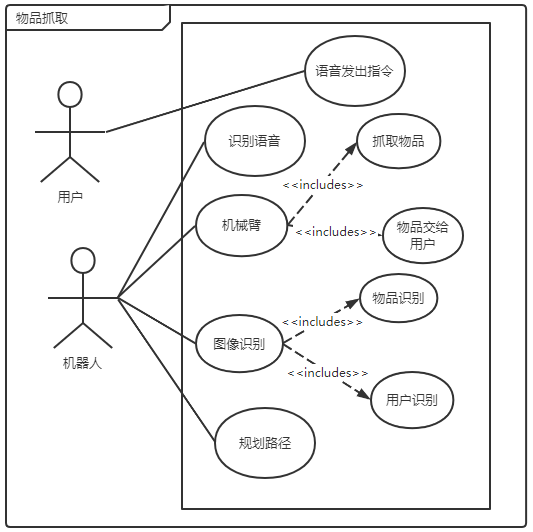
****

**图2 手动建图\_用例图**

## 用例二：物品抓取

**表3 物品抓取\_用例表**

|  |  |
| --- | --- |
| 主要参与者 | 机器人 |
| 前置条件 | 机器人完成了对该房间的建图  用户在图上分别标出所取物品和自己所在的大致位置 |
| 启动 | 用户要求机器人取某件物品 |
| 场景 | 用户开启机器人并登录；  用户选择进入“服务模式”；  用户选择所在房间的地图；  用户语音通知/输入机器人取具体物品；  机器人识别语音；  机器人移动到物品所在位置；  机器人通过摄像头识别物品位置，并进行抓取；  机器人移动到用户所在位置；  机器人通过摄像头识别用户位置，把物品交给用户；  机器人播报任务结束 |
| 异常情况分析 | 物品识别失败  没有移动到目标位置  识别物品失败 |
| 正常效果 | 机器人正常抓取物品，并交给用户 |
| 何时可用 | 第一次迭代 |
| 使用频率 | 高 |
| 次要参与者 | 用户 |

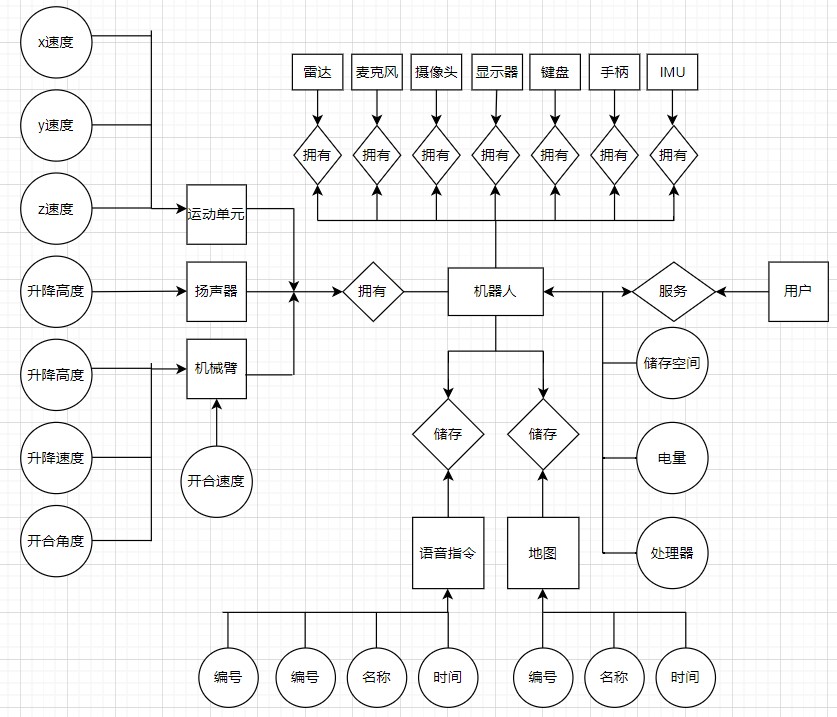
****

**图3 物品抓取用例图**

# 数据需求

## 数据实体及其关系

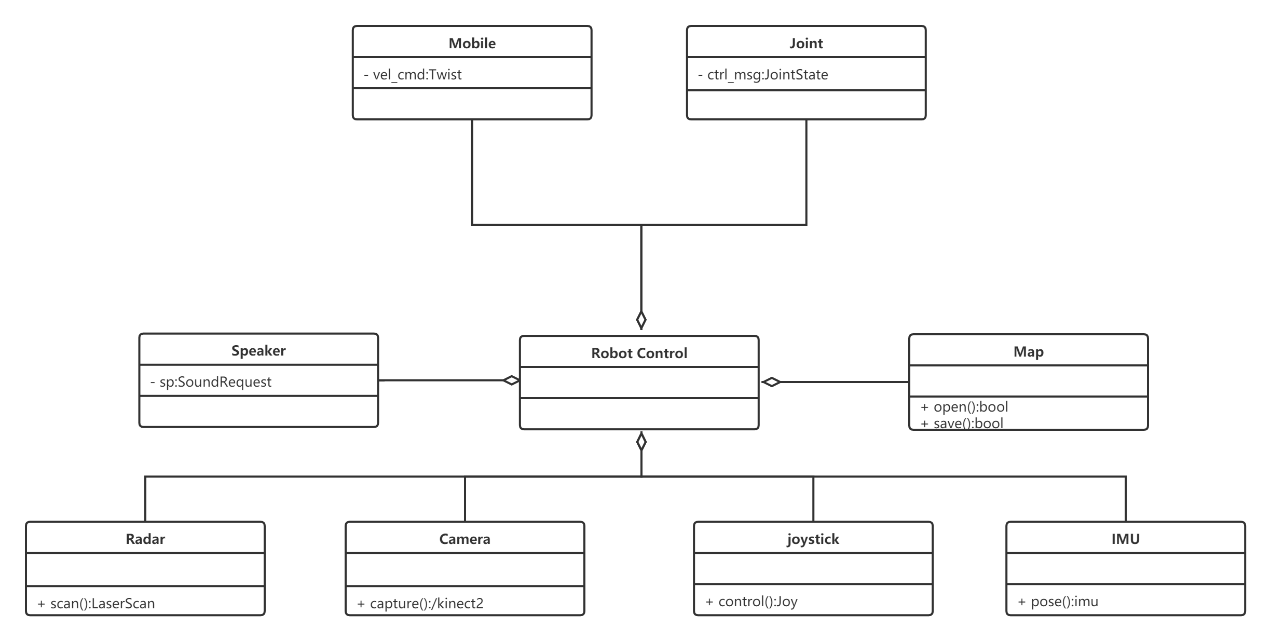
本机器人中所涉及到的实体主要有：用户、机器人、地图、语音指令、运动单元等、机械臂等机器人组件实体。实体的属性以及实体之间的联系如图5所示：

****

**图5 系统ER图**

## 类建模

经过分析讨论，本系统所建立的类模型包括：Robot Control类、Speaker类、Mobile类、Joint类、Map类、Command类、Rader类、Camera类、Microphone类以及IMU类。我们使用UML类图来描述具体的类的属性与方法以及类之间的关系，如图6所示：



**图6 系统类图**

# 功能需求

## 功能需求概述

本项目实现的是一个家庭服务机器人，需求主要分为机器人和用户界面两个部分，接下来按这两大部分并细化的介绍相应的功能需求。

机器人实现的主要功能是物品抓取并支持语音控制，随后详细介绍了相关的用户界面，以及系统的错误处理和可维护性。

## 物品抓取

机器人需要有物品抓取的能力，主要分为导航，取物放物，物品识别等功能。

### 导航

根据用户给出的物品和已经建立好的地图，寻找相应的位置，并自动规划路径。

### 取物/放物

机器人可以使用机械臂进行取物/放物，同时尽可能的降低对周围环境的影响。

### 物品识别

机器人可以根据环境中采集到的图片，识别物品类型，判断是否为所需的物品。

### 自动避障

在取物过程中，如果出现地图之外的障碍物，机器人有自动躲避障碍物并将其记录到地图上的功能。

## 语音交流

机器人支持语音交流，即可以通过识别用户给出的语音指令，做出相应的工作。

## 用户界面

### 启动界面

用户启动机器人后，启动界面自动弹出，同时用户可以进行对启动界面的操作。

### 选择界面

选择界面有三个按钮

1. “建图模式”：点击后可以进入建图界面，可以操纵机器人自动对房间进行建图，并且将建立的地图自动显示到屏幕上。
2. “服务模式”：点击后可以进入服务界面，用户可以自行选择需要的服务。
3. “标注模式”：点击后进入地图界面，用户可以自行对地图进行标注。
4. “退出”：点击后用户可以退出系统。

### 建图界面

1. 进入界面时，可以点击开始建图按钮操纵机器人进行建图。
2. 机器人建图时，显示“建图进行中”，并且可以显示建立的部分地图。
3. 当机器人建图完成后，系统会进行弹窗，提示用户是否保存地图。此时，用户可以选择是否保存地图，如果保存，则提示用户对此地图进行命名，如果不保存，那么退回到选择界面。

### 标注界面

用户在标注界面时首先需要选择需要标注的地图，选择后进入地图界面对地图进行标注。

### 地图界面

1. 用户进入地图界面后可以查看之前建立好的地图。
2. 右键点击相应地图可以修改地图名称或删除地图。
3. 左键点击相应地图可以查看地图详细信息。

### 维护界面

在维护界面下，用户可以选择进行“系统更新”或者“恢复出厂设置”，或者使机器人进行休眠。

## 异常处理

### 路径规划失败

当检测到障碍物过多或者因其他因素导致的路径规划算法失效，因此无法到达指定位置时，机器人会返回接受指令的位置，并通过用户的UI界面给出相应反馈。

### 物品识别失败

在物品抓取时需要进行物品识别，如果图像识别算法无法识别物品，或者没有识别到相应的物品时，机器人会返回接受指令的位置，并通过用户的UI界面给出相应反馈。

### 语音识别失败

当用户通过语音给机器人下达指令的时候，机器人会做出相应指令，如果语音识别识别或者识别出未定义的指令，机器人不会进行相应的工作，并通过用户的UI界面给出相应反馈。

### 用户界面异常

如果用户界面产生异常，会通过硬件的方式提示用户，此时用户需要根据相应的手册上的操作恢复UI界面或者恢复出场设置。

## 维护功能

### 恢复出厂设置

当用户错误的进行某些设定或者系统出现不可修复的bug 的时候，恢复出厂设置是不错的选择，因此，本项目设计了恢复出厂设置的API，通过调用该API，可以删除机器人的所有设置，即恢复出厂设置。

### 移动到新的环境

当移动到新的环境中时，旧的地图以及相应的数据需要删除，同时存入相应的新的数据，保证无用数据的随时删除，以此来保证整体数据空间足够用户使用。

# 非功能需求

## 性能指标

### 响应时间

机器人需要能对用户的指令作出及时的反应，本产品中机器人主要通过语音识别与GUI的方式与用户交互。对于GUI方式，机器人的反馈时间应在1秒以内；对于语音识别方式，机器人的反馈时间应在3秒以内。

### 功耗

机器人的功耗需求主要来自于电机运转、核心计算与屏幕显示。为了满足服务需求，机器人的总功耗应当能满足1分钟执行1条语音指令指导拾取物品时，持续运行60分钟的能力。

### 处理能力

机器人应当具备准确度高的语音识别功能，并根据功能作出图像识别、路径规划、物品拾取等功能。对于语音识别与图像识别部分，机器人应当能快速地完成计算并反馈结果；对于路径规划与物品拾取部分，机器人应当能准确无误地达成目标。

## 质量指标

### 系统可用性

机器人需要监听用户语音信息并执行指令，而图像识别、路径规划、物品拾取等功能只有在接收指令后才执行，因此语音系统服务在运行时间中拥有较大的占比，可以随时处理用户需求，具有较高的系统可用性。

### 可移植性

机器人采用ROS开发，而ROS的代码具备很强的可移植性，因此本项目同样能移植到其余平台上。

### 完整性

机器人预计实现语音识别、图像识别、路径规划、物品拾取等功能，并能与用户实现完整的GUI交互，具有较好的完整性。

### 效率

机器人应当以较高的效率运行。对于占比较大的语音监听模块，应当使其以一定小周期工作或停止，使其能监听到用户的指令，而不占用过多的功耗。同时，语音指令应在3秒内完成反馈。

### 健壮性

当指令无法执行时，机器人将会给出相应的反馈与提示，提高鲁棒性。同时，工程内部对机器人可能出现的各种错误情况都设置了可能的解决方案，使其更加健壮。

# 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 18.04 LTS；
* ROS Melodic Morenia；
* 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；在完成仿真开发迭代后，会将系统在真实的机器人环境下进行运行和测试。

## 开发环境

本系统开发所需的硬件环境为：

* 开发用计算机；
* 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 18.04 LTS；
* ROS Melodic Morenia；
* 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。