

智能抓取式机器人
需求规格说明书
SRS9
【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	对文档第一部分工作进行撰写
17373545	聂志捷	对文档第三、四部分工作进行撰写
17373513	吴仪周	对文档第二部分内容进行撰写，并对文档进行整理
17373541	薛晨祺	对文档第五部分工作进行撰写，并完成审稿
16061178	刘晓洁	对文档第六、七部分工作进行撰写

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	3.28	吴仪周	薛晨祺	软件需求规格说明书第一版

1. 范围

1.1 项目概述

开发背景: 本项目的开发主要是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, 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。

主要功能: 本项目待开发的嵌入式系统将主要用于单目标和多目标的识别、分类和抓取, 包含对不同形态的对象做出不同姿态的抓取, 同时还要求机器人能够与用户进行一些基础的语音交互从而应答用户的指令, 当然还包括对周围环境实时的建图, 以及抓取途中动态和静态避障等基础功能, 最终目标是开发一套具有较为完整功能体系的识别抓取机器人。

非功能性需求: 我们预期的嵌入式系统是健壮的, 能够在较短时间内对用户期望的目标进行识别和反应, 同时避障过程中也期望得到快速的反应, 即机器人内部的信息具有较高的更新率和反馈率。同时我们还需要信息具有一定的安全性, 对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。并且我们还希望指令设计尽可能简单, 易于用户进行操控。

应用场景: 我们产品的初步应用场景是作为家庭服务机器人的一个功能模块, 可以为用户提供一些简单的抓取运送功能, 同时我们的机器人也可以用于商场货运、物流分拣等领域, 并且可以根据要求进行二次编程从而统一规划线路。

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开发计划文档。用于描述小组项目的需求规格, 理清各个部分需要达到的标准等。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。

计划书的内容按照项目概述、业务需求、数据需求、功能需求、非功能需求、

用户界面需求和运行与开发环境来进行组织和编写,具体每个部分还进行了更细致的划分和补充,最终目的就是用于描述小组整个设计和开发 ROS 机器人软件工程项目各个部分的计划和要求,每个部分都由负责该部分的成员进行了认真的分析和调查,从而设计出更符合用户需要、更人性化的产品。

1.3 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
DFD	数据流图 (Data Flow Diagram)
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 版_开发手册——20181109

启智 ROS 机器人

2. 业务需求

2.1 业务环境

在家庭生活场所，用户下载相应平台通过语音操作可以对于机器人进行控制，对物品的搬运工作。机器人可以实时与用户进行交流，明确自己的任务目标。同时，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。

2.2 用户

本产品的用户分为普通家庭用户和管理者。

普通家庭用户可以对于机器人进行声控或手控要求，完成其抓捕任务。其具体表现为，登录进入界面之后，在初始界面选择不同的模式。

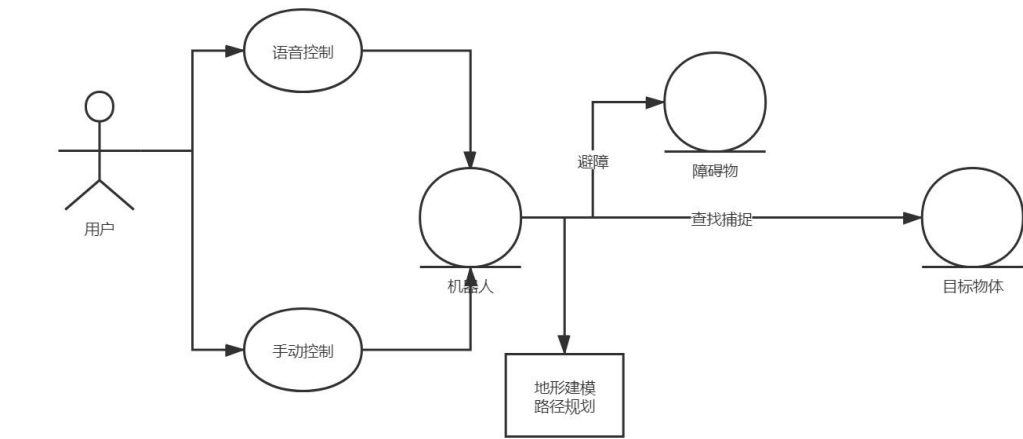
在手动操控状态下，机器人的运动轨迹由用户进行操作，操作界面将提供给用户方向键进行操作，此时机器人只进行路障检测，关闭地图建模和路径规划功能，当发现路障时进行警报并减速提醒用户，用户手动操作避过路障。当到达目的位置准备进行捕捉时，用户可从方向控制界面切换成目标捕捉界面，对机器人抓取臂进行操作，抓取到所需的目标物。

在自动模式下，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，过程中，用户可以随时语音或者手动选择停止机器人的抓捕任务。

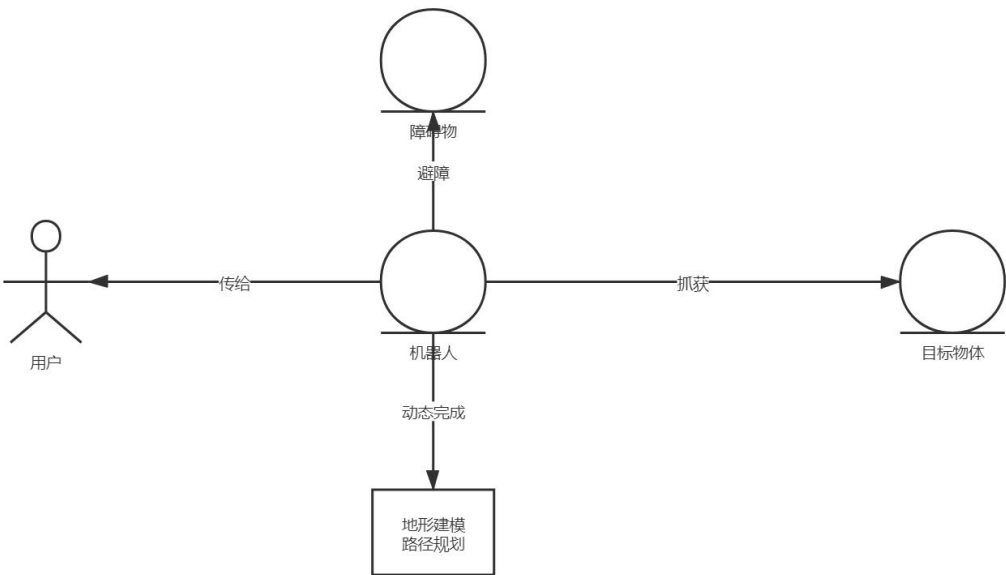
管理者可以对于机器人后台的路径进行进一步规划和调整，对于普通家庭用户的权限进行控制。在登录管理者界面后，可对机器人的基本情况进行操作，如清空后台数据，管理增加和删除普通用户信息等，同时管理者账号也可以对机器人进行如普通用户一样的操作。

2.3 主要业务

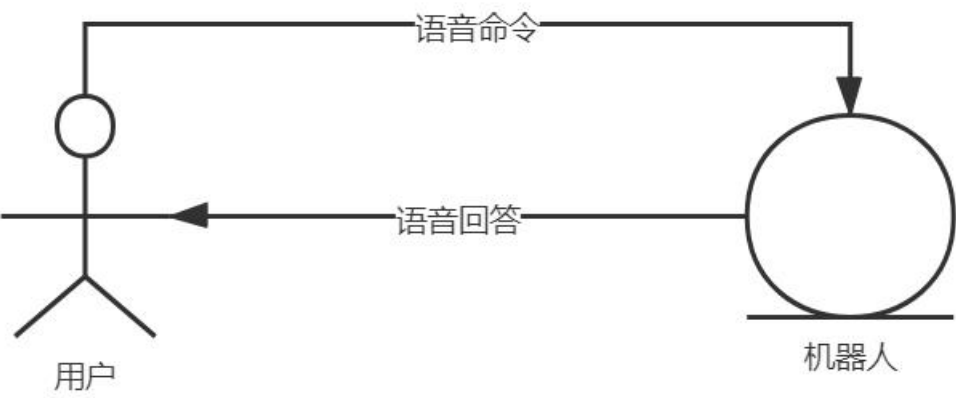
用户控制机器人取物品:



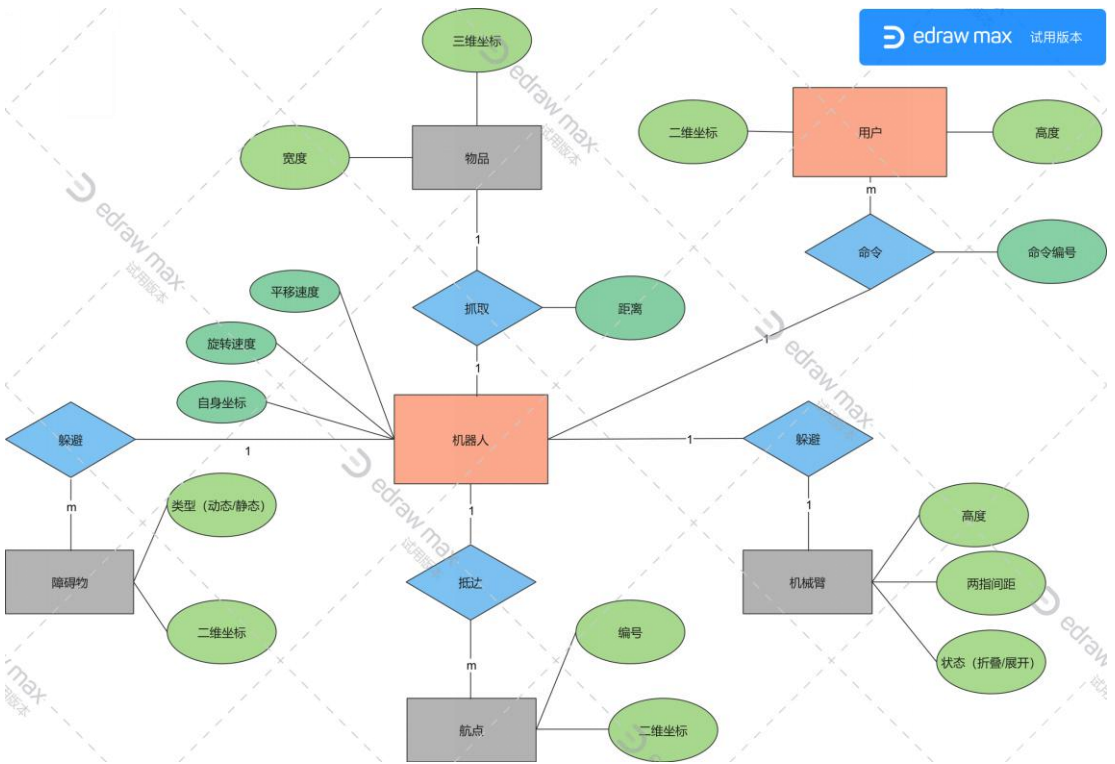
机器人拿到物品返回:

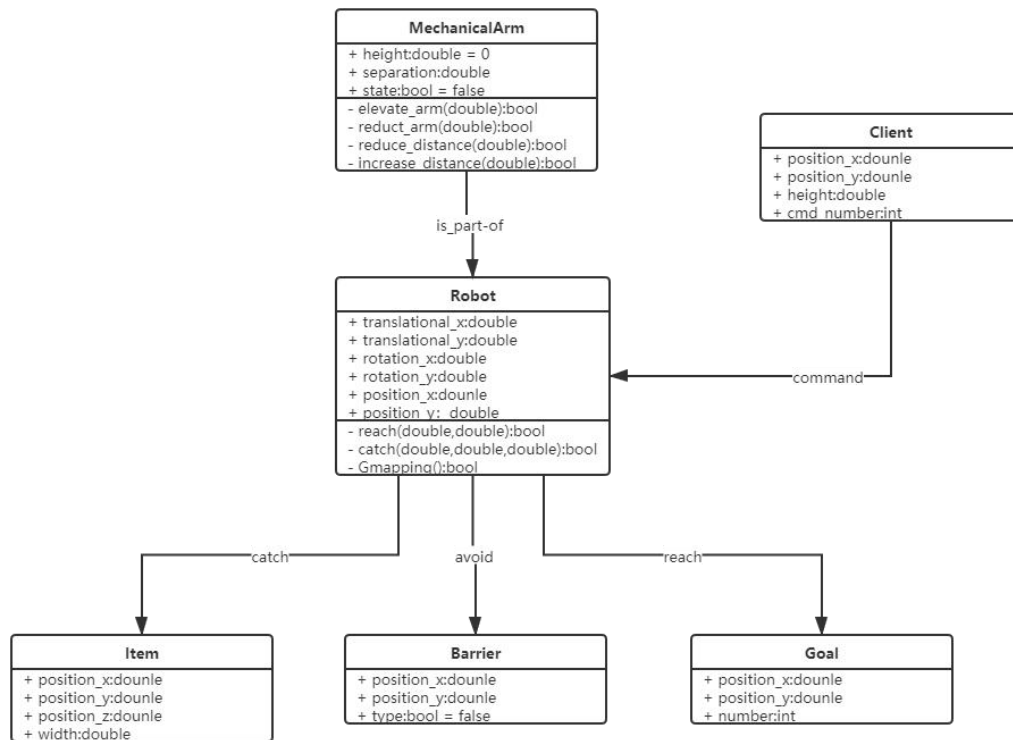


语音交互:



3. 数据需求





4. 功能需求

4.1 系统功能说明

4.1.1 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_msgs::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。

4.1.2 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。

4.1.3 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。

4.1.4 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;

4.1.5 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。

4.1.6 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_msgs::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

4.1.7 语音交互

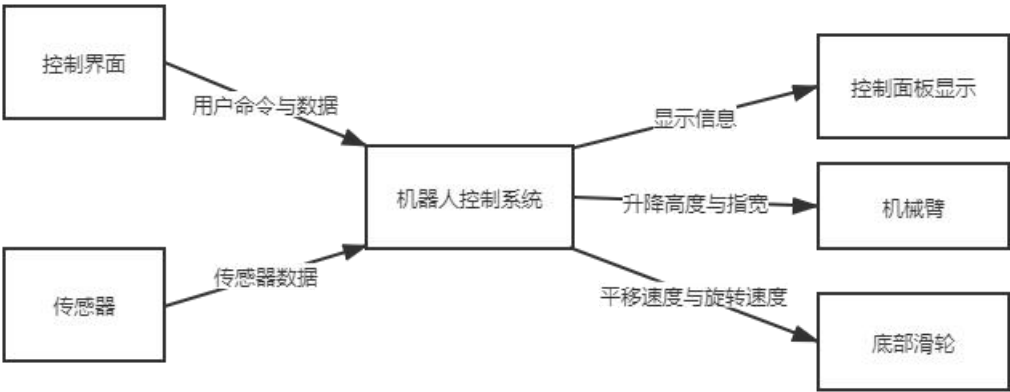
本功能的实现依靠科大讯飞的语音识别引擎和阵列麦克风实现。主要能够进行以下涉及语音交互的行为:

- (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。

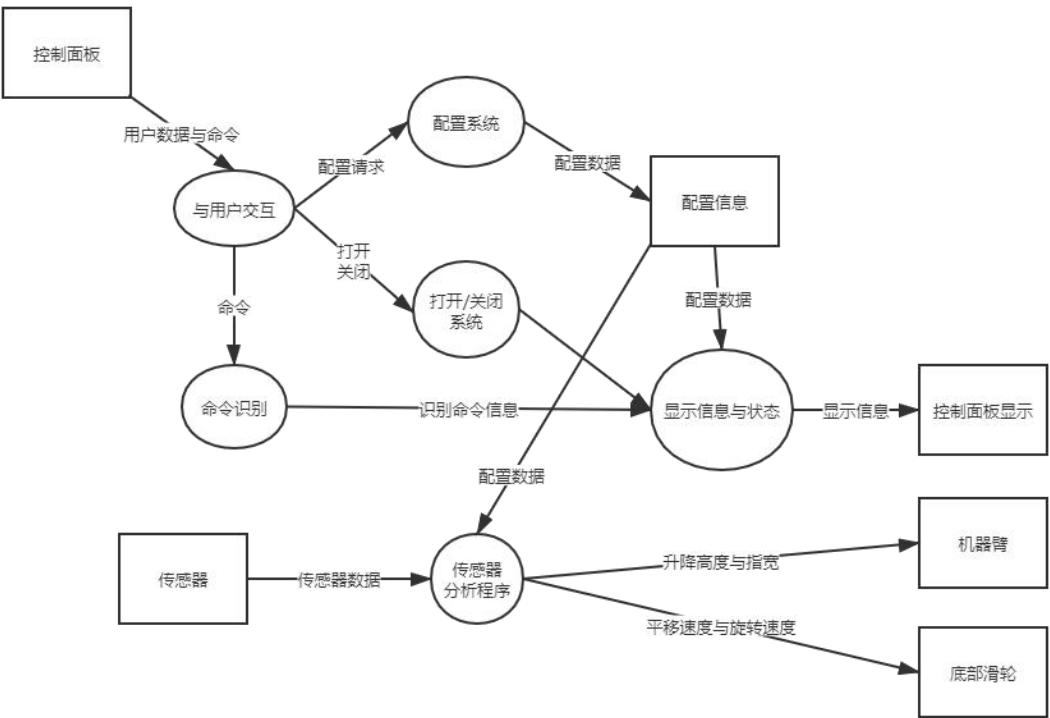
最终能够实现“特定地点物品抓取”和“递送饮料”等功能。

4.2 数据流图

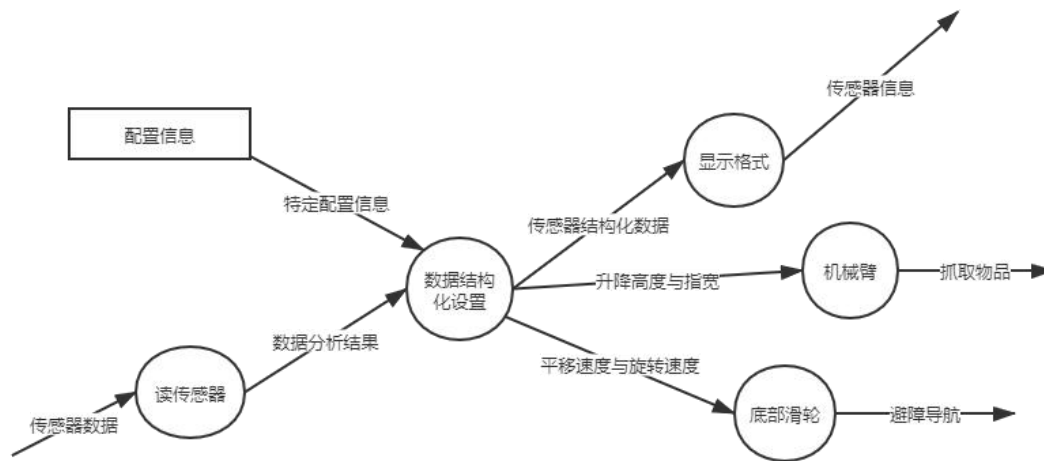
DFD Level0:



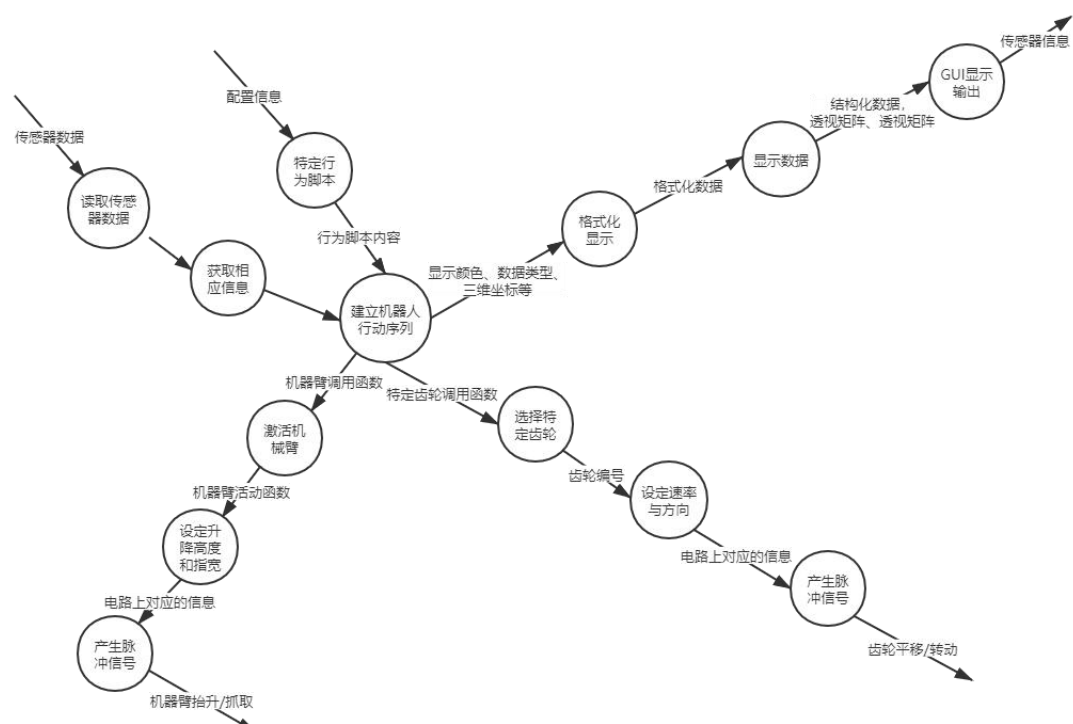
DFD Level1:



DFD Level2:



DFD Level3:



3. 机器人可以连续工作 4h 以上。

5.1.3 处理能力

1. 机器人内部的信息具有较高的更新率和反馈率。
2. 能够在较短时间内对用户期望的目标进行识别和反应。
3. 能够在避障过程得到快速的反应。
4. 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 $\leq 0.1\text{m}$ 。

5.2 质量属性

5.2.1 可用性

1. 易操作:控制指令的设计简单,用户可以在较短时间内学习使用方法,易于用户操作。
2. 容错能力:系统具有一定的容错能力和抗干扰能力。在非硬件故障或非通讯故障时,系统能保证正常运行。在 95% 的故障中,系统最多可以在 30s 内重启或解决故障。
3. 提供数据备份和恢复功能,若由于系统错误或其他原因导致系统数据丢失或被破坏时,可以及时的恢复和还原数据。

5.2.2 安全性

1. 对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。
2. 记录运行日志,对运行中的系统错误和用户关键操作信息进行记录,便于追踪。

5.2.3 可移植性

1. 本机器人只能运行在 Ubuntu16.04+ROS Kinetic 环境下,暂不支持移植。

5.2.4 可维护性与可扩展性

1. 保留历史各版本的源代码。
2. 将系统功能模块化,支持灵活开发,减少重复开发量。
3. 普通的修改请求或 bug 修复应在 1~2 天内完成,对于重大的需求或设计修改应在一周内完成。
4. 应有清晰的系统结构,命名规范,界面规范,提示和帮助信息规范。

6. 用户界面需求

1. 用户界面统一设计,保证界面文字、颜色、图案的一致性;符合美学标准。整洁、美观、错落有致。
2. 文字:字体使用适当,一般两到三种为宜;使用清晰易读方正的字体。
3. 颜色:颜色使用适当,颜色统一,色彩搭配简洁清楚,选择长期使用不易疲劳的颜色。遵循对比原则,深色背景使用浅色文字;浅色背景使用深色文字。
4. 图案:具有一致性,恰当,形象,使用户能够见图知意,大小合理。
5. 区域排列:整齐合理;一般的标签右对齐,编辑框左对齐。字符左对齐,数字右对齐。
6. 界面布局:合理的界面布局,保证平衡,屏幕不能拥挤,屏幕总体覆盖度最好不要超过 40%;而组合框中覆盖度不要超过 62% (Mayhew 1992 年试验结果)有效组合,在逻辑上关联的项目在屏幕上应当加以组合,以显示其关联性。反之任何项目之间毫不相关的项目应当分隔开。在项目集合间用间隔对其进行分组/或用方框也同样可做到这一点。保持习惯,屏幕组织自上而下、自左而右。

7. 运行与开发环境

7.1 运行环境

7.1.1 硬件环境：

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 50 μ s，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: kinetic 版本 ROS 系统, 基于 Ubuntu16.04

软件包:

软件包	说明	版本限制
启智 ROS 源代码		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		kinetic 版本
导航地图工具包		无
RPLIDAR ROS	激光雷达驱动	无

7.2 开发环境

7.2.1 硬件环境:

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：

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