**【项目名称】**

**需求规格说明书**

**【SRS203】**

**【version 1.0】**

分工说明

|  |  |  |
| --- | --- | --- |
| 小组名称 |  | |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 17373332 | 刘冬 | 分配任务，第一二三节编写及整改 |
| 17373344 | 杨华晟 | 第四，七节编写 |
| 17373342 | 邓力友 | 第三(2)，五节编写 |
| 77086002 | 萨扎尔 | 第六节编写 |

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 | 3.30 | 刘冬 | 杨华晟 | 第一版 |
| 2.1 | 4.22 | 刘冬 |  | 完善需求环境，修改 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 范围

## 项目概述

### 系统背景

目前，嵌入式系统已经深入到我们生活中的各个方面。小到录音笔，mp3，

大到汽车，飞机，上面都搭载着各式各样的嵌入式系统。嵌入式系统因为其适应硬件的独特软件系统，在功耗、成本、体积、可靠性、处理能力上都有着过人的表现而在各种工作类型清晰的设备上为人所青睐。

机器人同样也收到人们的欢迎。现在各式各样的机器人层出不穷，在不同的岗位上帮助人们完成着一些重复性高，危险性高，技术要求精细的工作。而机器人的自主导航是一大类机器人的核心问题。已经日趋成熟的扫地机器人，擦玻璃机器人，包括正在研究的自动驾驶汽车，对于路径的识别与规划都是重中之重。

在此基础上，我们将对有着嵌入式系统的机器人进行编码与设计。

### 主要功能性需求

我们设计的机器人的主要功能需求有：

* 机器人的主动控制 ；
* 机器人利用传感器实时建立环境地图 ；
* 机器人根据地图和自身的位置信息实现动态路径规划及导航控制；
* 物体的识别与自动抓取

### 非功能性需求

* 性能需求：响应时间短、短时间内吞吐量大、资源利用率高；
* 安全性：保密性、防泄漏、权限控制、防攻击；
* 可维护性与可扩展性：模块性、可复用性、易分析性；
* 可靠性：易恢复性、容错性、成熟性；
* 易用性：易学习性、易操作性、用户错误防御机制、用户界面美观；

### 应用场景

本机器人计划应用场景为仓库。快递物流量如此之大的今天，货物的分拣工作必不可少，但始终是由人类完成。我们希望设计出的机器人能够自主在待分拣货物堆中巡航，并最终通过对货物体积的大小的识别与自主的路径导航，在非人工干预下自主完成对货物的分拣。

### 文档用途

本文档的用途为明晰机器人的功能性与非功能性需求。对机器人进行初步的功能性与结构设计。

### 内容组织

本文档包含项目概述、文档概述、业务需求、数据需求、功能性需求、非功能性需求、用户界面需求与运行开发环境。

## 引用文档

SDP-软件开发计划203

启智ROS版\_开发手册\_20181109

启智ROS机器人

用户界面设计——有效的人机交互策略（第五版）\_201103\_0101版

# 业务需求

## 障碍识别与路径规划

管理员可以通过手动设定目的地，使得机器人去往指定地点。机器人在移动过程中需确认自己所在位置、根据当前位置与目的地进行实时规划、识别当前路径中的障碍物、躲避障碍物后再次进行路径规划。

## 目标物体识别与抓取

管理员在设定好机器人待抓取的物体之后，机器人需通过图像识别出该物体，并确认自己与该物体的相对位置。移动至可抓取的位置并将机械臂移动至抓取动作起始位置后，开始抓取动作。抓取后识别可放置物体地点，执行前述操作的逆操作，将物体放回。

## 货物自主识别分拣

管理员设定货物堆所在范围与分拣地点后，机器人通过自主巡航系统自主往返于两地之间，并根据识别出的货物大小将货物分拣至管理员所规定好的不同地点。用例图如下所示：

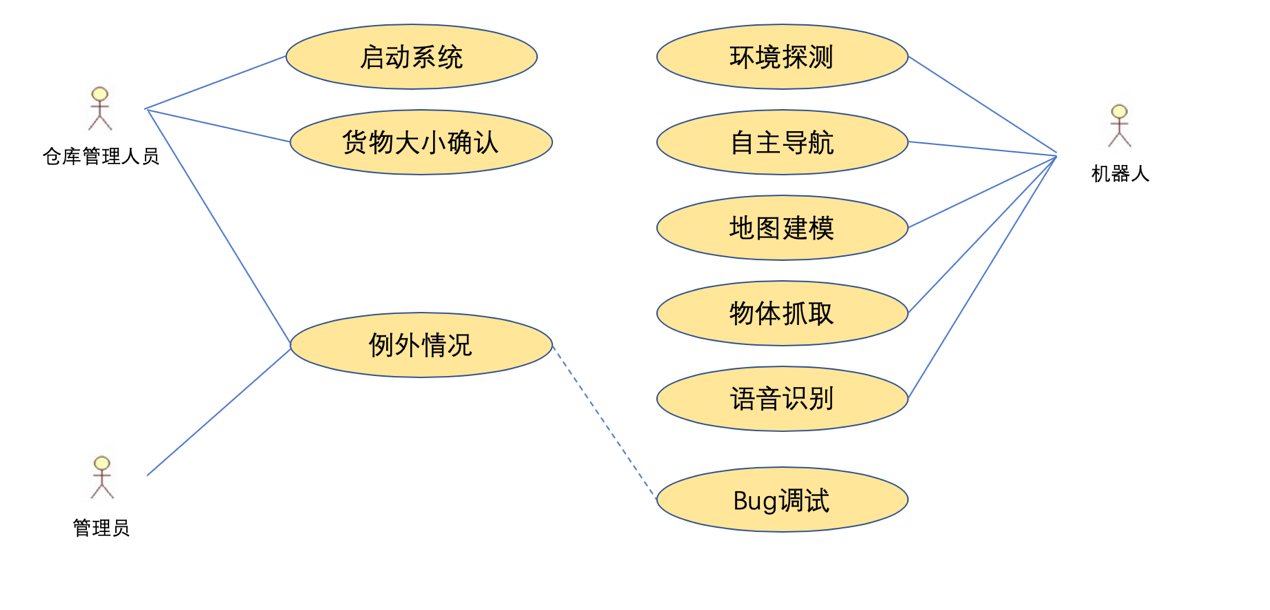


图 2-1 用例图

# 数据需求

ROS机器人的基本架构如下所示

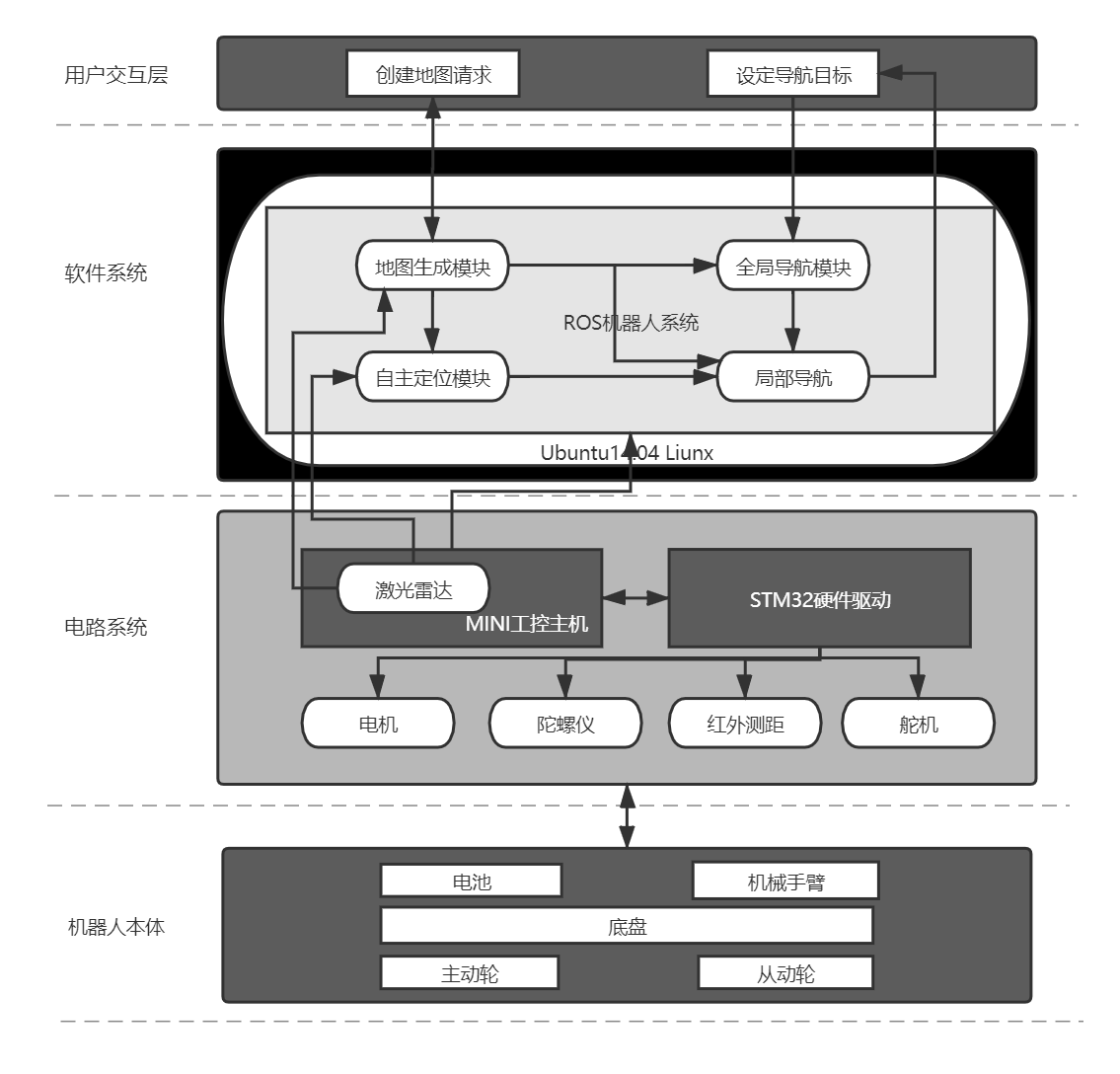


图3-1 机器人基本架构

根据机器人的基本框架及其已经封装好的功能，我们设计的类图如下：

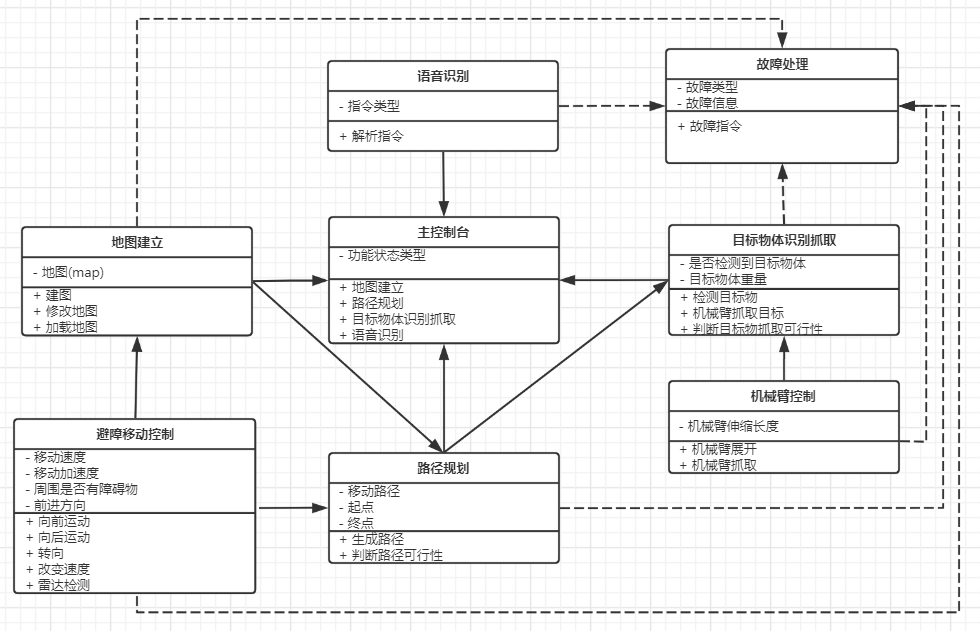


图 3-2 机器人类图设计

# 功能需求

## 语音识别

**主要参与者：**用户

**目标：** 识别用户的语音输入指令，从中解析出目标物体和目标地点（可无）

**前置条件：**机器人正常

**启动：**用户操作正确

**场景：**

1. 用户发出指令
2. 机器人收到并解析指令
3. 指令解析正确则执行，否则提示用户重新输入指令

**优先级：**中

**何时可用：**用户发出语音指令时

**使用频率：**高

**次要参与者：**无

## 地图建立

**主要参与者：**管理员

**目标：** 对指定区域进行扫描并建立地图

**前置条件：**机器人开机且可以正常运行

**启动：** 机器人在指定区域运动并扫描周边环境，建立地图

**场景：**

1. 管理员下达地图建立指令，并指定区域。
2. 机器人对指定区域进行遍历，并不断用激光雷达扫描周边环境
3. 在扫描过程中建立三维地图，对检测到的障碍物位置进行标记
4. 将地图存储在数据库中

**优先级：**高

**何时可用：**机器人转移到未知场所时

**使用频率：**低

**次要参与者：**激光雷达

## 路径规划

**主要参与者：**用户

**目标：** 打开/关闭机器人

**前置条件：**地图建立完毕

**启动：**用户操作正确

**场景：**

1. 从用户指令中解析出目标地与目标物
2. 对目标地进行路径规划
3. 机器人开始运动
4. 运动过程中检测障碍，若出现新的障碍则重新进行路径规划并修改地图
5. 到达指定目标点

**优先级：**高

**何时可用：**机器人关闭/开启时

**使用频率：**中

**次要参与者：**机器人运动模块，激光雷达

## 目标物体识别抓取

**主要参与者：**用户

**目标：** 机器人在路径规划结束后成功识别出用户指定物体的位置并抓取目标物

**前置条件：**路径规划成功

**启动：**路径规划完毕后启动

**场景：**

1. 从用户指令中解析出目标地与目标物

2. 对目标地进行路径规划

3. 到达目标地

4. 检测目标物体

5. 完成抓取

**优先级：**高

**何时可用：**机器人开机时

**使用频率：**中

**次要参与者：**机械臂

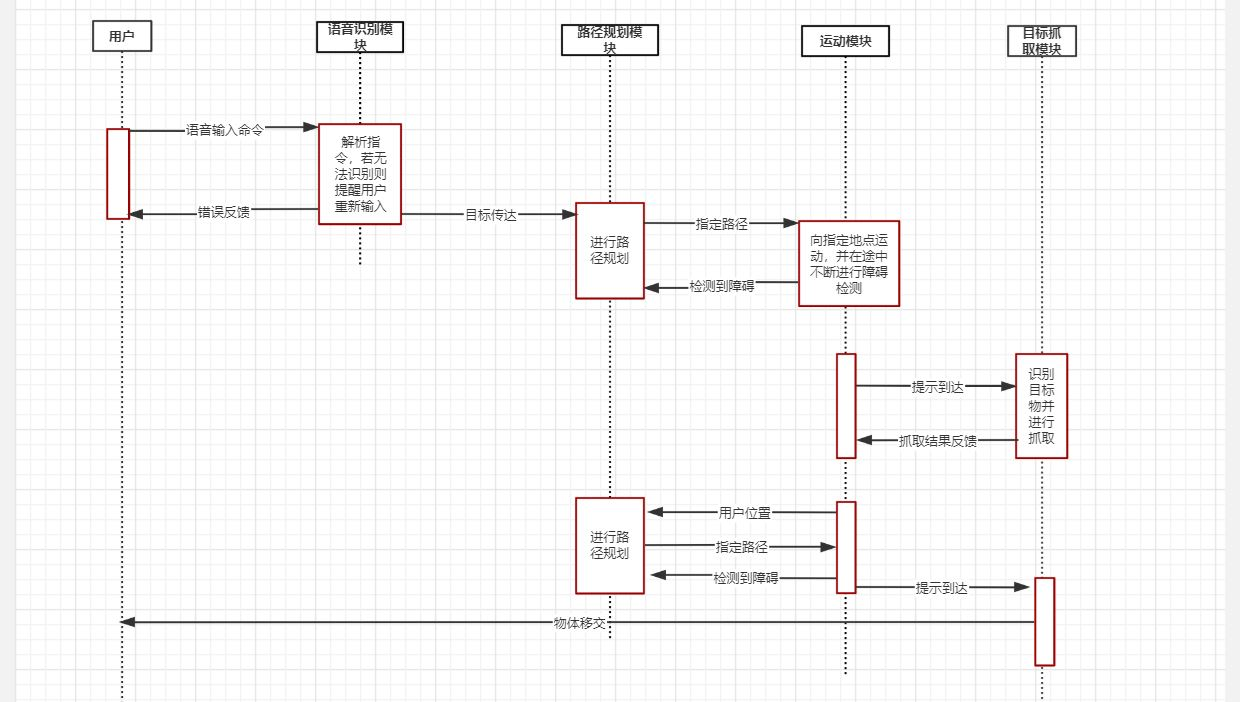


图 4-1 功能实现时序图

# 非功能需求

## 性能指标

### 响应时间

* 机器人的主动控制响应时间应在0.5-1s内；
* 当机器人移动过程中在其激光雷达范围内突然检测到有障碍物接近，则应在1-1.5s内减速至停止；

### 功耗

* 在机器人行驶过程中，应当保持在0.15m/s左右的速度匀速行驶，在没有障碍物的情形下减少变速运动；

### 处理能力

* 对于已知起点和终点位置的简单环境，机器人路径规划应在100s之内完成起点至终点的行进；对于未知的复杂环境，机器人应在环境感知过程中缓慢接近终点，预计在400s之内完成行进；
* 当目标物出现在机器人感知识别的范围内时，机器人应在1-3s内选择合适的路径动态接近目标物；
* 当机器人抓取目标物时，应该确保抓取目标物行进的过程中不掉落，能够适应最少0.2m/s的行进速度；

## 质量属性

### 系统可用性

* 用户能够正常操作控制系统，确保起点、终点的合理规划，如果出现起点和终点不可到达的情况，系统应及时提醒用户，并且不进行任何动作；
* 系统操作应当简洁，确保每个用户学习后能够成功上手；

### 系统可移植性

* 机器人每项功能代码应分为独立模块，代码耦合度不应过大；
* 在确保代码的独立性后，新增功能时保证原有代码的可用性；
* 提供适当的接口，保证BUG的修复不影响其他功能；

### 系统安全性

* 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；
* 启智ROS机器人的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。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；
* 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；
* 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；

### 系统效率

* 在用户设定好起点和终点时，系统应当在5s内规划出一条最优路径，并能在1s内从静止加速到0.1m/s，即将抵达终点时，应保证机器人在0.2m左右开始减速，同时准备伸展机械臂，在抵达终点的同时，能在5s以内快速抓取目标物，返回起点；

# 用户界面需求

用户界面的基本要求为：

* 界面元素：  
  用户界面中的所有元素应简单明了，并且不包含任何错误。 每个按钮的含义都很直观。 界面的颜色样式简单易用。
* 用户角色：

界面面向的用户角色具有一定的计算机基础，用户使用软件具有一定的熟练度。

* 操作方法：  
  通过使用鼠标和键盘与界面元素互动。用户可以随时使用操作手册。
* 全部功能在用户界面的按钮可以进行想要的操作而且可达的。例如目的地址，启动等。
* 能够实时反映软件运行的状态（正常或异常），显示机器人用到的传感器和算法产生的结果。
* 具有一定的安全性，当用户危险操作时，能够弹出警告窗口并且提醒用户做出的应急举措。

系统界面的草案如图6-1、6-2：

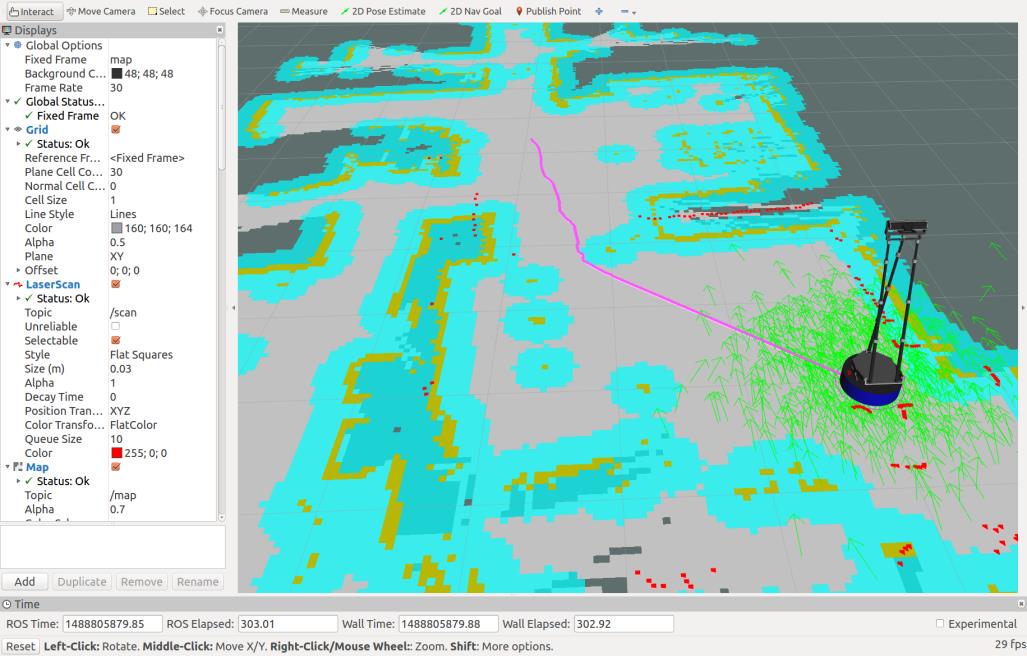


图6- 1

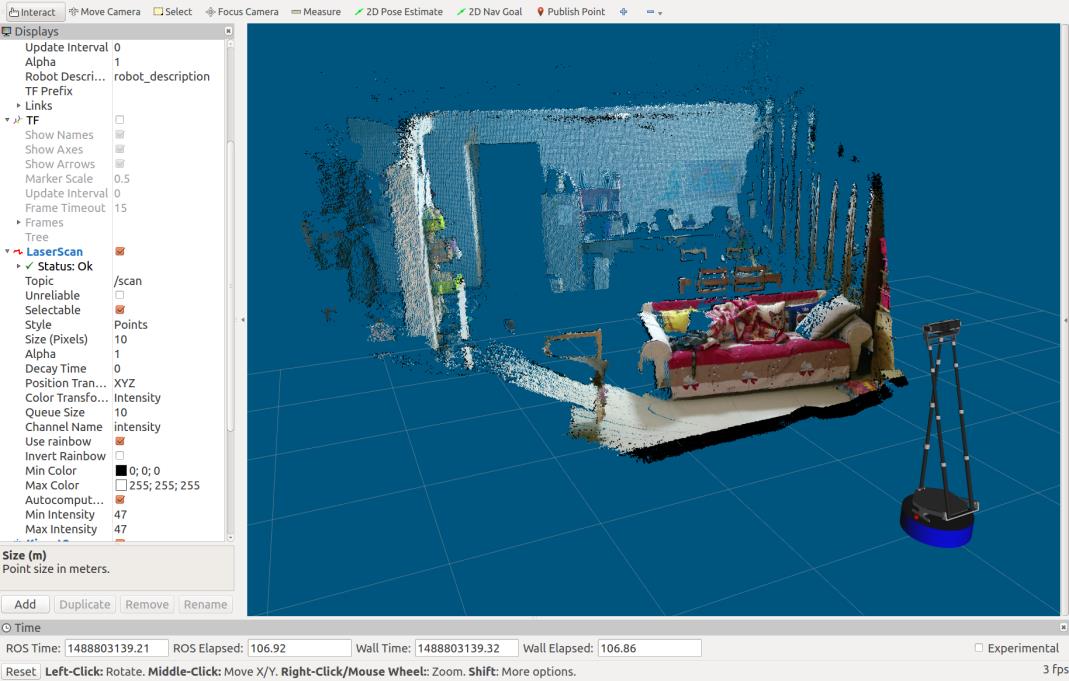


图6- 2

使用ROS里最常用的图形化显示界面Rviz来进行操作。点击 Rviz 界面上方工具栏条里的“2D Pos Estimate”按钮，然后再点击 Rviz 的地图 里，现实机器人所处的位置。这时，会出现一个绿色大箭头，代表的是机器人在初始位置的朝向。现在用户通过鼠标可以控制机器人的初始位置。

设置好机器人的初始位置后，可以开始为机器人指定导航的目标地点。

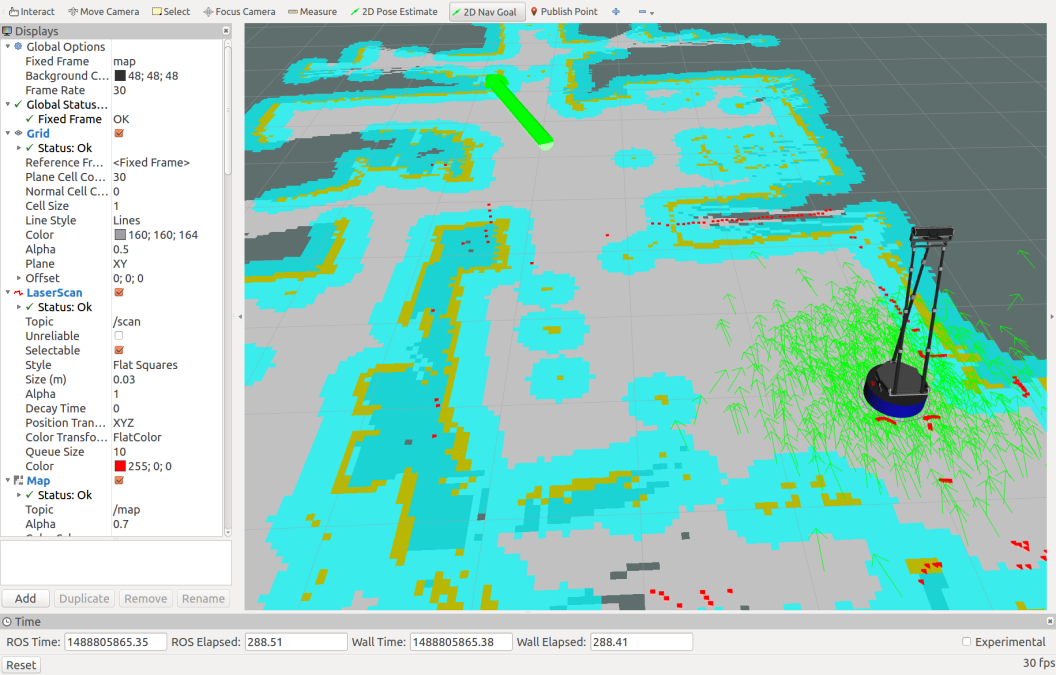


图6- 3

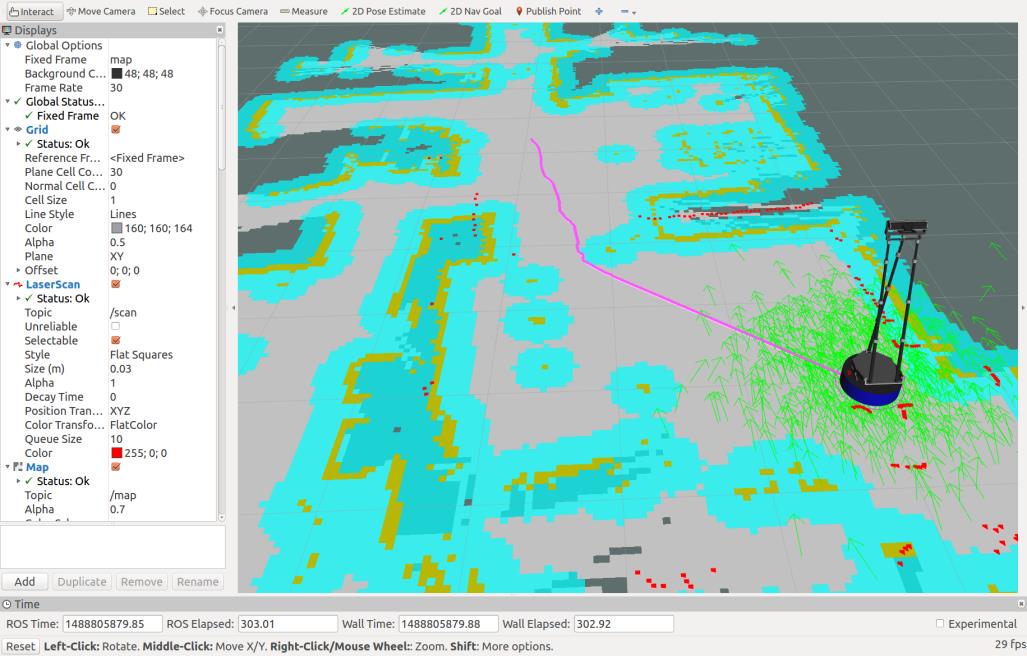


图6- 4

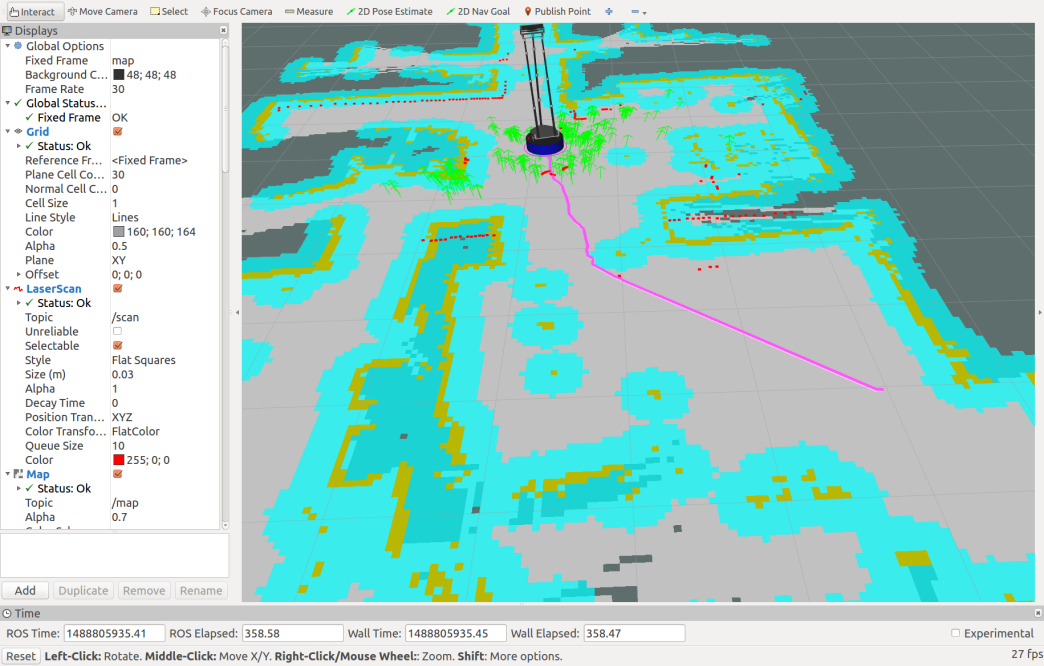


图6- 5

# 运行与开发环境

## 运行环境

**硬件环境：**

* 启智ROS机器人

**软件环境：**

* Ubuntu 16.04 / Ubuntu14.04
* ROS Kinetic(Ubuntu 16.04) / Ros Indigo (Ubuntu 14.04)

## 软件环境

**硬件环境：**

* 个人PC
* 最低配置:
* 操作系统: Ubuntu 14.04 or 16.04
* 处理器: Dual core from Intel or better
* 内存: 4 GB RAM
* 显卡: nVidia 320M or higher, or Radeon HD 2400 or higher, or Intel HD 3000 or higher
* 网络: 宽带互联网连接
* 存储空间: 需要 15 GB 可用空间

**软件环境：**

* Ubuntu 16.04 / Ubuntu14.04
* ROS Kinetic(Ubuntu 16.04) / Ros Indigo (Ubuntu 14.04)

C++(/Python)开发环境