**启智ROS机器人开发计划**

**需求规格说明书**

**SPD203**

**1.0.1**

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 |
| 16061095 | 王贺正 | 范围及业务需求部分 |
| 16061132 | 杨光 | 功能需求部分 |
| 15271124 | 高则晗 | 运行与开发环境部分 |
| 16061148 | 单勇 | 数据需求部分 |
| 16061139 | 阳韵非 | 非功能需求部分 |

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 | 3.31 | 王贺正、杨光、高则晗、单勇、阳韵非 | 王贺正 | 初版 |
| 1.0.1 | 4.1 | 王贺正 | 王贺正 | 部分修改 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

.

目 录

[1. 范围 1](#_Toc264820566)

[1.1 项目概述 1](#_Toc264820567)

[1.2 文档概述 1](#_Toc264820568)

[1.3 术语和缩略词 1](#_Toc264820569)

[1.4 引用文档 2](#_Toc264820570)

[2. 业务需求 2](#_Toc264820571)

[3. 功能需求 3](#_Toc264820572)

[4. 数据需求 5](#_Toc264820573)

[5. 非功能需求 6](#_Toc264820574)

[6. 运行与开发环境 7](#_Toc264820575)

[6.1 运行环境 7](#_Toc264820576)

[6.2 软件环境 7](#_Toc264820577)

[6.3 用户界面需求 7](#_Toc264820578)

# 范围

## 项目概述

本项目基于启智ROS机器人，开发可操纵机器人行动的嵌入式系统。主要功能有：提供可手动操作机器人行动的方法，实现机器人读取地图后的路径规划功能，实现机器人自动扫描周边环境生成地图并寻路，实现机器人使用其机械臂对物品进行夹取。同时要保证机器人在恶劣环境中的可用性，用户使用过程中的安全性，简化使用界面保证产品的易用性，以及控制机器人的电量消耗，保证机器人对于未来更多功能的可拓展性。

用户可以使用开发完成后的产品的机械臂夹取功能完成车间生产线上固定物品的取用，也可结合使用多种功能使产品成为智能家居的一部分，帮助用户完成各项家务活动。

## 文档概述

本文档用于介绍机油教驻泰伦研究队所进行的启智ROS机器人开发计划的需求分析，包括业务需求、功能需求、数据需求和非功能需求。

## 术语和缩略词

表格 1 术语表

|  |  |  |
| --- | --- | --- |
| 编号 | 术语或缩略词 | 解释 |
| 1 | USB | Universal Serial Bus, 通用串行总线 |
| 2 | CPP | C++语言 |
| 3 | ROS | Robot Operating System机器人操作系统 |
| 4 | GUI | Graphic User Interface图形用户界面 |
| 5 | IDE | Integrated Development Environment集成开发环境 |
| 6 | Wi-Fi | 一个创建于IEEE 802.11标准的无线局域网技术 |
| 7 | git | 一个开源分布式版本控制系统 |

## 引用文档

表格 2 引用文档表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 标题 | 版本 | 发行日期 |
| 1 | 启智ROS机器人开发手册 | V1.1.0 |  |
| 2 | 启智ROS机器人开发计划 | V1.1.1 | 2019.3.31 |

# 业务需求

该项目的产品主要用于室内使用，需要在倾斜程度不大于15度的水平面上进行工作。理想的工作环境为家庭、办公室、工厂或仓库等场所。要求能够自动或是通过手动操作实现寻路和移动功能，并且通过手动操作可利用机械臂实现对物体的抓取，其流程图如下：

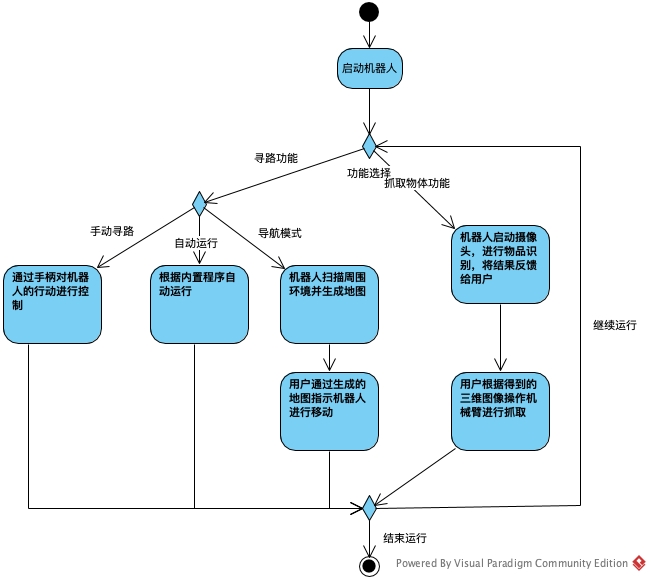


图 1 业务需求活动图

# 功能需求

3.1系统用例模型及简要说明

机器人系统将采用活动图方式来展示具体操作流程。

原因：对于机器人的驾驶操作以及机械抓取操作步骤非常明确，可拆分。且在进行一项操作时着重使用机器人的某几个模块，在操作完成之前极少调用其他模块，因此不需要在用例模型图中体现出扩展性。

3.2对系统用户的描述

使用者：机器人操作员，机器人开发者，机器人附带传感器。

使用功能描述：

功能1：机器人操作员可以通过手柄直接控制机器人。

功能2：机器人开发者可以编写软件使机器人自动按照代码运行。

功能3：机器人操作员打开机器人附带传感器，使得雷达传感器扫描整个运行空间，生成三维地图，机器人可按照地图进行自动寻路方式运行。

功能4：机器人操作员可按照机器人生成的三维地图，操纵机器人附带的机械臂进行抓取操作。

3.3功能1用例概述

主要参与者：机器人操作员。

目标：通过外部设备，直接操纵机器人的行走模块使得机器人改变所处位置。

前置条件：机器人所附带计算机中已安装ROS等相应操作软件。

启动需求：机器人操作员试图移动机器人的当前所处位置。

场景流程：

1.机器人操作员将控制计算机安装到机器人上，连接USB口并连接操控手柄。

2.打开机器人底盘面板的USB电源开关。

3.启动Ubuntu终端，并输入roslaunch wpb\_home\_bringup js\_ctrl.launch指令。

4.待终端上显示出速度信息显示行时，可以拨动手柄摇杆。

5.打开机器人底盘面板上的红色急停开关，此时可以正常移动。

优先级：最高。

何时可用：开机后随时。

使用频率：极高。

可能的次要参与者：机器人自带探测雷达。

3.4功能2用例概述

主要参与者：机器人开发者。

目标：通过直接编写程序，使得机器人不用外部操控也能自主行走。

前置条件：同功能1。

启动需求：机器人开发者试图调试机器人运行代码。

场景流程：

1.机器人开发者打开RoboWare Studio，新建ROS包，输入名字与依赖项。

2.机器人开发者于src目录中新建CPP源文件，选择“加入到新的可执行文件”选项。

3.机器人开发者编写代码并编译。

4.启动机器人核心节点，确认打开电源，输入rosrun指令，命令机器人运动。

优先级：低。

何时可用：机器人开发者编写程序后。

使用频率：低。

可能的次要参与者：机器人操作员。

3.5功能3用例概述

主要参与者：机器人开发者，机器人操作员，机器人自带传感器。

目标：通过雷达传感器对周围空间的扫描，获取空间地图信息并保存，在地图的基础上进行导航移动。

前置条件：同功能1，并且完成功能1。

启动需求：机器人操作员试图让机器人能够在确定空间中自主寻路移动。

场景流程：

1.启动机器人以及激光雷达，并启动Rviz平台。

2.操作机器人在确定空间场地内移动，保存地图。

3.在可视化地图文件中，规划机器人的行走路径。

4.机器人自动在路径中行走。

优先级：高。

何时可用：机器人开发者完成对确定空间的地图获取，并在操作员完成路径规划后。

使用频率：中。

可能的次要参与者：无。

3.6功能4用例概述

主要参与者：机器人操作员，机器人自带传感器。

目标：通过机器人自带传感器对于桌面物体的识别后，操纵机器人抓取物品。

前置条件：同功能1，并且对于机器人机械臂调整完成。

启动需求：机器人操作员试图让机器人识别并抓取桌面上的某个特定物件。

场景流程：

1.启动机器人头部的摄像头，检测三维空间的物体。

2.收到三维图像后，机器人操作员操控机械臂，进行抓取操作。

优先级：中。

何时可用：机器人操作员完成三维空间物品识别后，并且调节摄像机摆放角度等软硬件调试工作后。

使用频率：中。

可能的次要参与者：机器人开发者。

# 数据需求

主要数据实体：控制面板，雷达传感器，机器人

关系：控制面板 机器人，

雷达传感器收集数据 控制面板 机器人，

机器人 正常移动，机器人 抓取物品。

数据项：控制面板（配置信息，空间三维图像，规划路径图像）

雷达传感器（空间地图信息）

# 非功能需求

**5.1、功能性**

• 适合性：机器人的机械臂能够适应基本机械臂操作，红外线扫描系统能够获取地图。

• 准确性：机器人操作系统响应时间小于0.2s，寻路系统对于终点的误差不超过20cm，机械臂对物体的识别率在98%以上。

• 依从性：依据启智机器人开发手册。

• 安全性：机器人将拒绝除制定操作设备以外的任意设备给出的指令。

**5.2、可靠性**

• 成熟性：机器人故障率小于0.1%。

• 容错性：在机器人接受到异常指令后会停止当前动作。

• 易恢复性：在出现异常问题后，机器人系统可以恢复出厂设置，重新录入数据。

**5.3、易用性**

• 易理解性：用户在操作过程中面对的是GUI界面。

• 易学习性：用户通过说明文档通过程序控件操作即可。

• 易操作性：GUI界面上可进行机器人操作模式的选择，录入地图，机械臂启动等操作均可通过按键完成。

**5.4、效率**

• 时间特性：录入指令时间小于0.5s。实时操作的响应时间小于0.2s。系统恢复出厂设置时间小于10s。

• 资源特性：工作电压在23.1v-29.4v 之间。

**5.5、维护性**

• 易分析性：程序将纪录机器人的每一次行为轨迹。

• 易改变性：根据异常轨迹对相应代码进行修改。

• 稳定性：出现异常行为的概率小于0.2%。

• 易测试性：测试数据均有GUI界面输入。

**5.6、可移植性**

• 适应性：需要制定安装环境。

• 易安装性：安装包自动生成软件环境，控件一键录入指令。

• 遵循性：遵循本项目的小组的开发文档。

• 可替换性：无。

# 运行与开发环境

## 运行环境

系统运行的硬件环境为有Kinect摄像头、单片机运动控制器外设、扬声器、麦克风、机械支架等的树莓派。树莓派3B+是一台卡片式小型计算机，它有一个处理器系统芯片（SoC）BCM2837B0，内嵌四核运行于1.4GHz主频的Cortex-A53处理器内核，支持64位ARMv8指令集。它还有1GB低功耗低电压的LPDDR2 DRAM，并且支持以太网，Wi-Fi，USB，串口等对外通信。

系统运行的软件环境为ARM版本的Linux发行版Ubuntu操作系统，以及其上的ROS元操作系统。Ubuntu是常用的Linux发行版，它为该系统提供了一套稳定易开发的运行环境。ROS (Robot Operating System, 机器人操作系统) 提供一系列程序库和工具以帮助软件开发者创建机器人应用软件。它提供了硬件抽象、设备驱动、函数库、可视化工具、消息传递和软件包管理等诸多功能。ROS遵循BSD开源许可协议。

## 开发环境

系统开发所需的硬件环境为上位机PC、与下位机通信的连接器以及机器人开发套件。我们预计使用以太网或Wi-Fi，在下位机树莓派上搭建SSH和VNC服务器来实现连接。

系统开发所需的软件环境主要为集成开发环境（IDE）RoboWare Studio，RoboWare Studio是一个对ROS开发非常友好的IDE，它支持自动配置ROS编译选项，部署代码到下位机，git版本控制等功能。

## 用户界面需求

用户界面包括机器人功能选择界面，以及各个功能对应的操作界面。具体而言，（第三部分中介绍的）功能一需要提供一个接受外部手柄信号的接口及机器人行进速度信息。功能二不需要提供任何接口或用户界面。功能三需要为用户提供生成后的地图以及通过地图接受目的地的功能。功能四需要为用户提供摄像头得到的三维图像以及接受外部手柄信号的接口。