**浇花机器人——温室管家**

**需求规格说明书**

**SRS-03**

**V2.1**分工说明

|  |  |  |
| --- | --- | --- |
| 小组名称 | 北航优质小组 | |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 6 | Kun | 主编，小组讨论，功能需求 |
| 2 | Hao7un | 审核，小组讨论，业务需求 |
| 9 | LTT | 小组讨论，功能需求 |
| 1 | Le | 小组讨论，运行与开发环境，数据需求，范围 |
| 5 | Mxode | 小组讨论，非功能需求，用户界面需求 |

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V1.0 | 2024.3.23 | Kun | Hao7un | 软件需求规格说明初稿 |
| V1.1 | 2024.4.25 | Hao7un | Kun | 更新需求规格 |
| V1.2 | 2024.4.27 | Le | Kun | 更新范围、数据需求（软件类图）和运行环境； |
| V1.5 | 2024.5.4 | Kun | Hao7un | 根据软件设计调整需求规格：   1. 将“缺水识别”功能改为“缺水判定”（含水量曲线）；   2. 将动态避障视作导航的基本功能。 |
| V1.6 | 2024.5.5 | Kun | Hao7un | 1. 将“缺水巡检”用例改为“巡检维护”用例； 2. 将自动建图和手动建图分离；   3. 加入等待状态服务，添加用例图； |
| V2.0 | 2024.6.20 | Kun | Hao7un | 文档一致性迭代：1. 调整时序图中模块的名称与ROS包名一致；2. 修改时序图部分逻辑错误 |
| V2.1 | 2024.6.21 | Le | Kun | 更新E-R图 |

# 范围

## 项目概述

* + 1. **背景**

当前，温室种植业面临着劳动力成本上升和管理效率低下的双重挑战，这限制了产业的规模化和高效率生产。同时，精准农业的发展要求更高的作物管理精度和环境适应能力。从技术领域上看，近年来随着嵌入式开发的成熟以及人工智能领域的高速发展，将机器人与温室种植业结合的交叉应用逐渐成为可能。

如今，市面上已经有一定数量的温室种植机器人投入使用，但现有温室种植机器人仍存在“自主导航与环境感知能力不足”，“无法实时监测作物缺水情况”，“难以精确识别作物并自动浇灌”等问题，这些问题极大程度上限制了温室管理智能化的发展。

本项目的目标是基于嵌入式系统开发一款“温室管家”浇花机器人，集成精确建图、动态避障、交互界面等技术来实现自动浇灌功能，降低人工成本并提高温室管理的智能化水平，特别适用于针对番茄等温室作物的精心照料。

* + 1. **主要功能**

本机器人的主要功能将涵盖温室种植业中的主要需求，具体而言包括：

* 可调节的工作模式，分别对应以下6种用例：
  + 自动建图模式：用户无需操作机器人运动，机器人自动对温室扫描建图。
  + 手动建图模式：用户可以使用手柄控制机器人运动扫描温室以进行建图。
  + 巡检维护模式：机器人巡检查看是否有花盆位置发生移动，并维护植物数据库。
  + 定点浇水模式：用户可以指定一个或多个特定的花盆位置，机器人将导航到这些位置并执行浇水任务，确保特定的花卉得到充足的水分。
  + 自动巡检浇水模式：在执行巡检路线时，机器人自动检测各花盆的水分状态。一旦发现缺水的花盆，立即进行浇水，并记录相关信息。
  + 等待服务模式：包含导航，设置机器人在温室中的初始位置等功能，便于用户灵活便捷地使用机器人。
* 动态避障功能：在执行上述模式时，机器人具备环境感知能力，能够识别并动态避开路径中的障碍物，确保安全地完成制定任务。
* 系统急停功能：除了机器人自带的硬件急停开关，用户也可以通过用户界面上的急停按钮实现软件急停，进一步保证作业安全性。
* 人机交互界面：机器人配备用户友好的人机交互界面，使用户可以方便地进行更换工作模式、查询花卉状态等操作。
  + 1. **非功能性需求**

·可靠性与稳定性等特性：为了保证高效、安全地完成用户的需求，浇花机器人需要具有可靠性、稳定性等特性，为用户提供良好的使用体验。

·异常处理功能：机器人拥有完备的异常处理功能，能够在遇到故障或异常情况时，自动采取恢复措施或将问题上报给用户，确保系统能够稳定运行。

* + 1. **应用场景**

**·**种植产业温室：适用于不同规模的温室花卉种植，提供个性化的浇水方案。

**·**科研机构：为研究人员提供精准控制的植物水分条件，助力农业科研。

**·**城市农业项目：适用于城市屋顶花园、垂直农业等新型农业形态。

## 文档概述

### 用途

本软件需求规格说明书主要用于结合当前农业背景明确项目的业务需求和用例，并在此基础上确定软件具体需要实现的主要功能与数据，便于团队成员有条理地进行协作开发。

### 内容组织

* 项目概要
* 业务需求
* 数据需求
* 功能需求
* 非功能需求
* 用户界面需求
* 运行与开发环境

## 术语和缩略词

### 术语

Ubuntu 18.04 LTS：一个以桌面应用为主的Linux发行版操作系统

ROS Melodic： 机器人操作系统

C++: 一种支持面向对象和底层操作的高性能编程语言

Python：一种简洁易懂，功能强大的解释型高级编程语言

Rviz：机器人可视化开发平台

Gazebo：机器人仿真开发平台

Visual Studio Code：代码编辑与开发工具

GitLab：基于Git的集成软件开发平台

### 缩略词

表1 缩略词/全称表

| **缩略词** | **全称** |
| --- | --- |
| ROS | Robot Operating System |
| WBS | Work Breakdown Structure |
| SDP | Software Development Plan |
| SRS | Software Requirement Specification |
| SDD | Software Design Description |
| STP | Software Test Plan |
| STR | Software Test Report |

## 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] 机械工业出版社，《ROS机器人开发实践》，2018-5.1

# 业务需求

随着机器人技术和人工智能的迅速发展和深度融合，现代农业正在经历一场前所未有的革新，智能农业在种植业中的应用日渐普及且日趋精细化。尤其是在温室栽培中，先进的自动化技术已经不仅仅是概念设想，而是实实在在地走进了日常的花卉灌溉管理实践。

温室环境为植物提供了理想的生长条件，然而其精细化管理往往需要大量的人力和专业知识投入。智能农业技术的应用为此带来了革命性的转变，自动化灌溉系统依托嵌入式技术和人工智能算法，机器人系统可以根据收集到的数据对花卉进行自动灌溉，确保花卉在不同生长阶段得到最适宜的水分，从而提升花卉品质，并最大限度地减少水资源浪费。

现代智能农业解决方案正在逐步引领传统种植业迈向自动化、信息化和智能化的新发展阶段。本项目的智能机器人浇水系统具备显著的创新性与实用性，具备广阔的业务前景和潜在的经济效益。

## 业务概述

**业务环境概述**

本项目的适用场景主要包括但不限于商业温室、科研机构的研究设施及公共花园内的密闭空间。系统设计针对这些稳定环境下的花卉灌溉作业，从而降低人力成本和提高灌溉效率。本系统配备直观易用的人机交互界面，便于各类用户快速掌握操作。

**核心用户群体分析**

1. **实验室研究人员**：借助本机器人系统，在实验室环境下可精确控制对植物的浇水操作，以满足不同实验条件下对水分管理的需求。
2. **花卉种植管理人员**：在温室大棚等封闭种植区域部署本系统，能实现花卉灌溉工作的自动化，显著节约人力成本，从而增强企业的经济效益。

## 用例分析

本机器人系统的总用例图如下图所示，本系统总共包含**自动建图**、**手动建图**、**巡检维护**、**定点浇水**、**自动巡检浇水**、**等待服务**共六个用例。

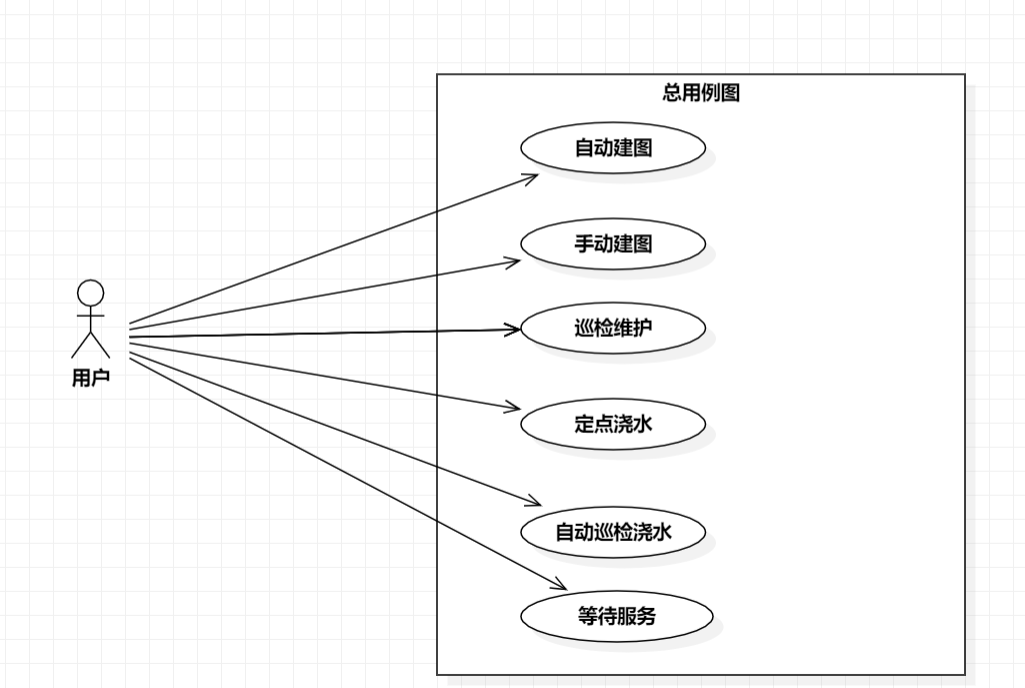


图1 总用例图

### 用例一：自动建图

表2 自动建图用例表

|  |  |
| --- | --- |
| 目标条件 | 用户启动自动建图命令后，机器人使用空间搜索算法自动为室内建图 |
| 前置条件 | 机器人启动并且机器人正常工作 |
| 触发条件 | 用户启动自动建图模式 |
| 场景 | 1. 用户启动机器人电源 2. 用户选择进入“自动建图模式” 3. 机器人进行反馈“开始建图” 4. 机器人将已建立的地图展示在用户交互界面上 5. 机器人执行完空间搜索算法后自动结束建图 6. 机器人将建图结果进行保存，同时在用户界面显示“已完成建图” |
| 异常情况分析 | 1. 建图过程中机器人因电源耗尽、机械故障等因素导致建图终止 2. 机器人因模型部分与房间内实体发生碰撞 3. 机器人无法搜索温室所有区域导致建图不完全 |
| 何时可用 | 第二次迭代 |
| 使用频率 | 中 |

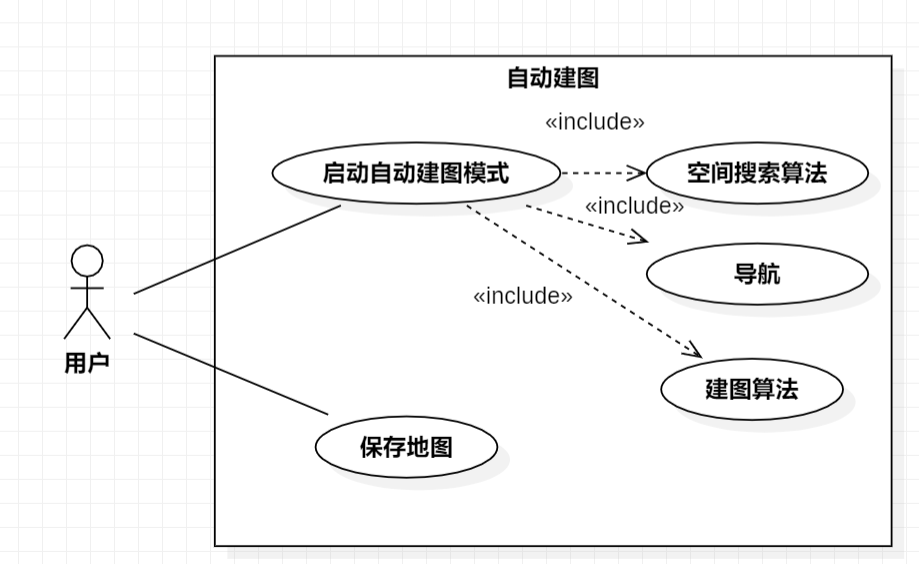


图2 用例图\_自动建图

### 用例二：手动建图

表3 手动建图用例表

|  |  |
| --- | --- |
| 目标条件 | 用户控制机器人对使用场景进行扫描并建立地图 |
| 前置条件 | 机器人启动并且机器人正常工作 |
| 触发条件 | 用户要求机器人启动手动建图模式对房间进行建模 |
| 场景 | 1. 用户启动机器人电源 2. 用户选择进入“手动建图模式” 3. 机器人进行反馈“开始建图” 4. 机器人将已建立的地图展示在用户交互界面上 5. 用户通过手柄（键盘）操作机器人移动，对温室进行扫描 6. 用户根据当前机器人建图结果判断是否已经对房间完成建图 7. 用户点击建图结束 8. 机器人将建图结果进行保存，同时在用户界面显示“已完成建图” 9. 用户点击确认，返回到交互界面 |
| 异常情况分析 | 1. 建图过程中机器人因电源耗尽、机械故障等因素导致建图终止 2. 机器人因模型部分与房间内实体发生碰撞 |
| 何时可用 | 第一次迭代 |
| 使用频率 | 高 |

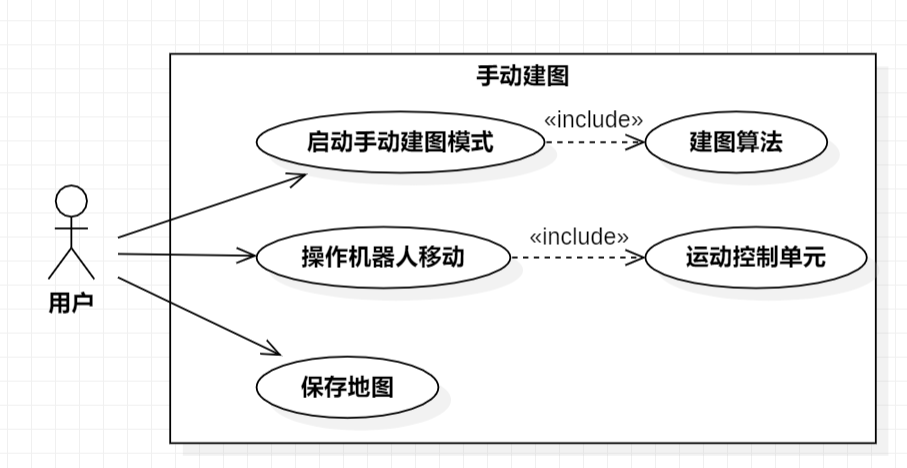


图3 用例图\_手动建图

### 用例三：巡检维护模式

表4 巡检维护用例表

|  |  |
| --- | --- |
| 目标条件 | 机器人巡检中检测花盆位置是否发生移动，并从数据库中删除已经缺失的花盆 |
| 前置条件 | 1. 机器人启动，并且能够正常工作 2. 机器人已对房间完成建图 3. 机器人自身位置已知 |
| 触发条件 | 用户启动巡检维护模式 |
| 场景 | 1. 用户开启机器人 2. 用户选择进入“巡检维护模式” 3. 机器人按照巡检路线进行移动，导航至每一个花盆的前方，同时使用摄像头对沿线花盆进行识别 4. 若未检测到花盆则从数据库中删除该花盆信息。 5. 机器人巡检到指定路经的终点，交互界面显示“巡检维护已完成” 6. 机器人返回至初始地点 7. 交互界面返回至上一级界面 |
| 异常情况分析 | 1. 花盆并未改变位置但识别失败导致数据库误删 2. 因地图发生改变，目标地点无法到达 |
| 何时可用 | 第二次迭代 |
| 使用频率 | 低 |

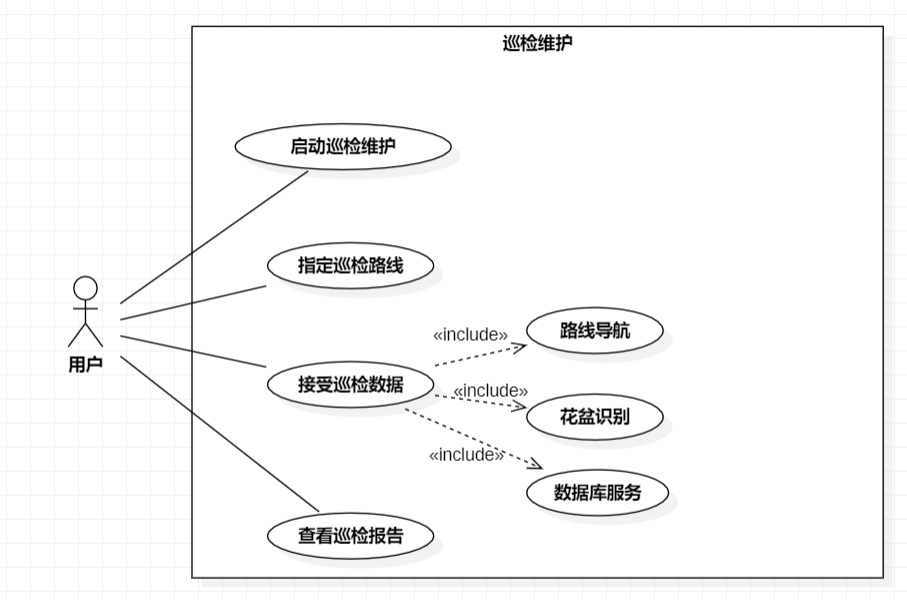


图4 用例图\_巡检维护

### 用例四：定点浇水模式

表5 定点浇水用例表

|  |  |
| --- | --- |
| 目标条件 | 用户选择待浇水的目标花盆，机器人依次对花盆浇水 |
| 前置条件 | 1. 机器人启动，并且能够正常工作 2. 机器人已对房间完成建图 3. 机器人自身位置已知 |
| 触发条件 | 用户要求机器人按照指定路线执行定点浇水模式 |
| 场景 | 1. 用户开启机器人 2. 用户选择进入“定点浇水模式” 3. 用户在交互界面选择若干个目标花盆 4. 用户在交互界面指定机器人的初始位置 5. 机器人根据选择的花盆位置进行路径规划 6. 机器人根据生成的路径前往指定的花盆 7. 机器人通过摄像头对花盆进行识别，并操作机械臂进行浇水，数据库更新该花盆的上一次浇水的时间 8. 机器人完成最后一个花盆的浇水，交互界面显示“指定位置浇水已完成” 9. 机器人返回至初始地点 10. 交互界面返回上一级界面 |
| 异常情况分析 | 1. 花盆识别失败 2. 因地图发生改变，目标地点无法到达 3. 机械臂并未对准花盆导致浇水偏移 |
| 何时可用 | 第一次迭代 |
| 使用频率 | 高 |

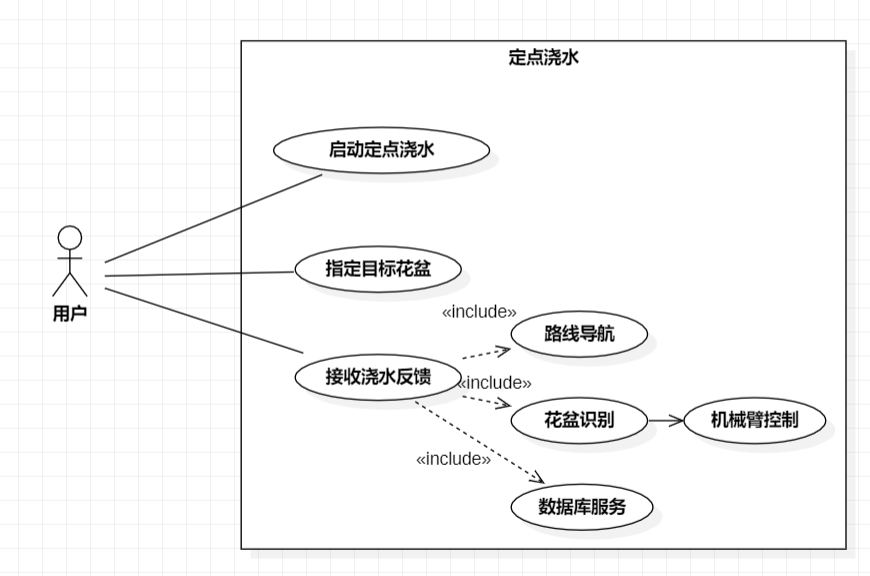


图5 用例图\_定点浇水

### 用例五：自动巡检浇水模式

表6 自动巡检浇水用例表

|  |  |
| --- | --- |
| 目标条件 | 机器人根据每盆花上一次浇水的时间模拟每盆花的含水量，并对其中缺水的花盆浇水。 |
| 前置条件 | 1. 机器人启动，并且能够正常工作 2. 机器人已对房间完成建图 |
| 触发条件 | 用户启动自动巡检浇水模式 |
| 场景 | 1. 用户开启机器人 2. 用户选择进入“自动巡检浇水模式” 3. 机器人获取数据库中所有花盆上一次浇水的时间，并使用含水量变化曲线计算当前含水量，并判断是否缺水。 4. 如果花盆缺水，机器人导航至该花盆前方，操作机械臂对花盆进行浇水，更新数据库 5. 机器人完成所有缺水花盆的浇水后，交互界面显示“巡检自动浇水已完成” 6. 机器人返回至初始地点 7. 交互界面返回至上一级界面 |
| 异常情况分析 | 1. 花盆识别失败 2. 因地图发生改变，目标地点无法到达 3. 机械臂并未对准花盆导致浇水偏移 |
| 何时可用 | 第一次迭代 |
| 使用频率 | 高 |

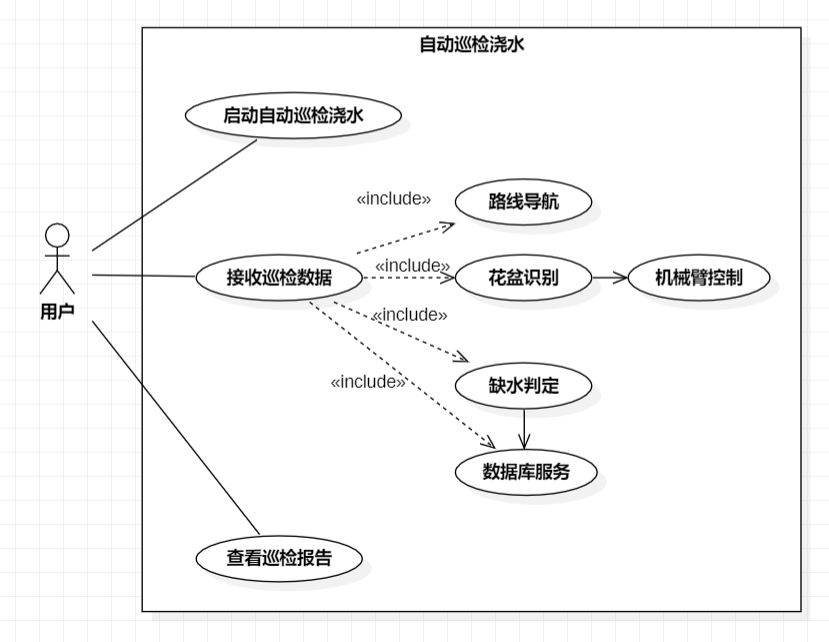


图6 用例图\_自动巡检浇水

### 用例六：等待模式

表7 等待模式用例表

|  |  |
| --- | --- |
| 目标条件 | 这是机器人闲置状态下（即没有执行前5种用例时）可以调用的功能。该模式下集成了导航，手动设置机器人位置，清除地图等三个功能。 |
| 前置条件 | 1. 机器人启动，并且能够正常工作 2. 机器人已对房间完成建图 |
| 触发条件 | 机器人已经启动并且处于闲置等待状态（没有执行前五种用例）。 |
| 场景 | 1. 用户开启机器人，默认进入“等待模式” 2. 用户选择“导航”，“手动设置机器人位置”，“清除地图”等三个功能。 3. 若选择导航，用户在前端地图上选择航点后，机器人将导航至该航点 4. 若选择手动设置位置，则用户在前端地图上选择机器人的位置和姿态后，将重新设置机器人在世界参考系下的位置和姿态 5. 若选择清除地图，则机器人将重启地图服务，并删除原有的地图文件 |
| 异常情况分析 | 1. 导航功能：航点不可达 |
| 何时可用 | 第一次迭代 |
| 使用频率 | 高 |

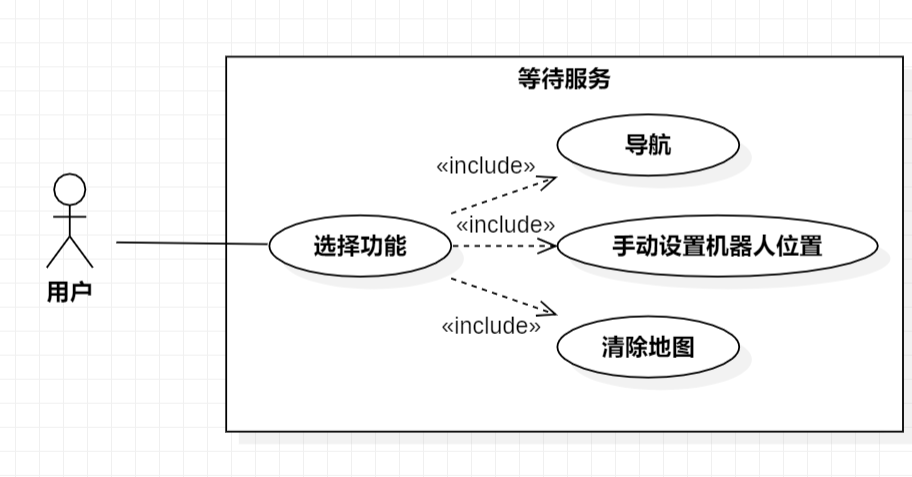
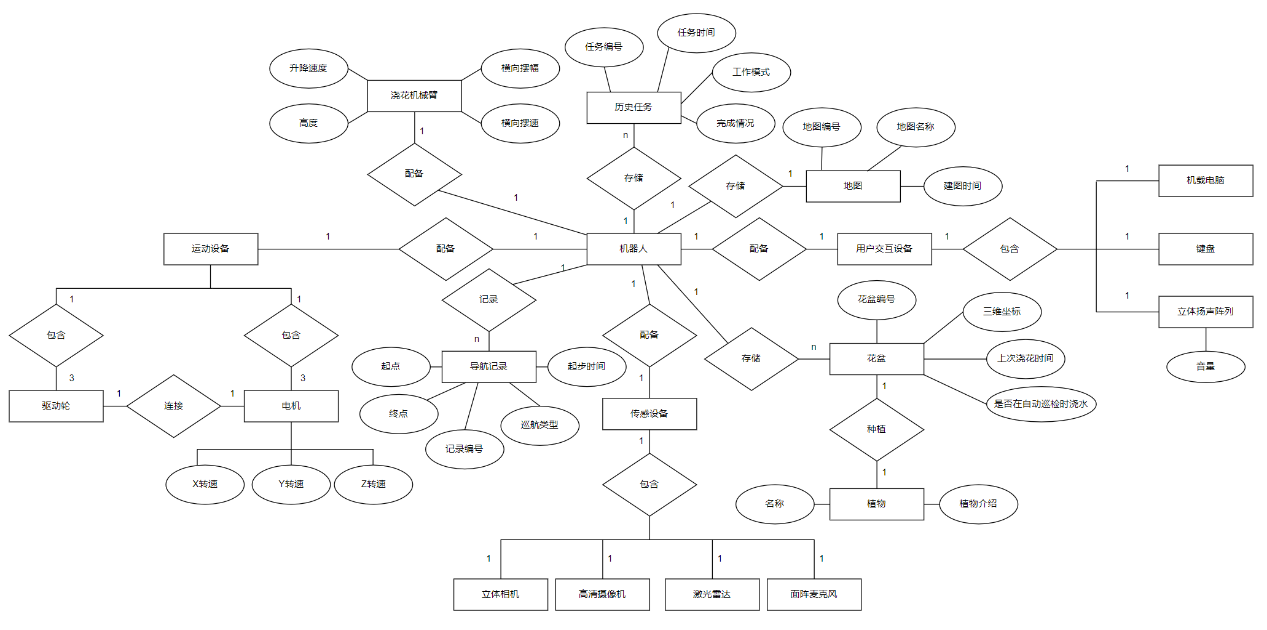


图7 用例图\_等待服务

# 数据需求

## 实体-关系图

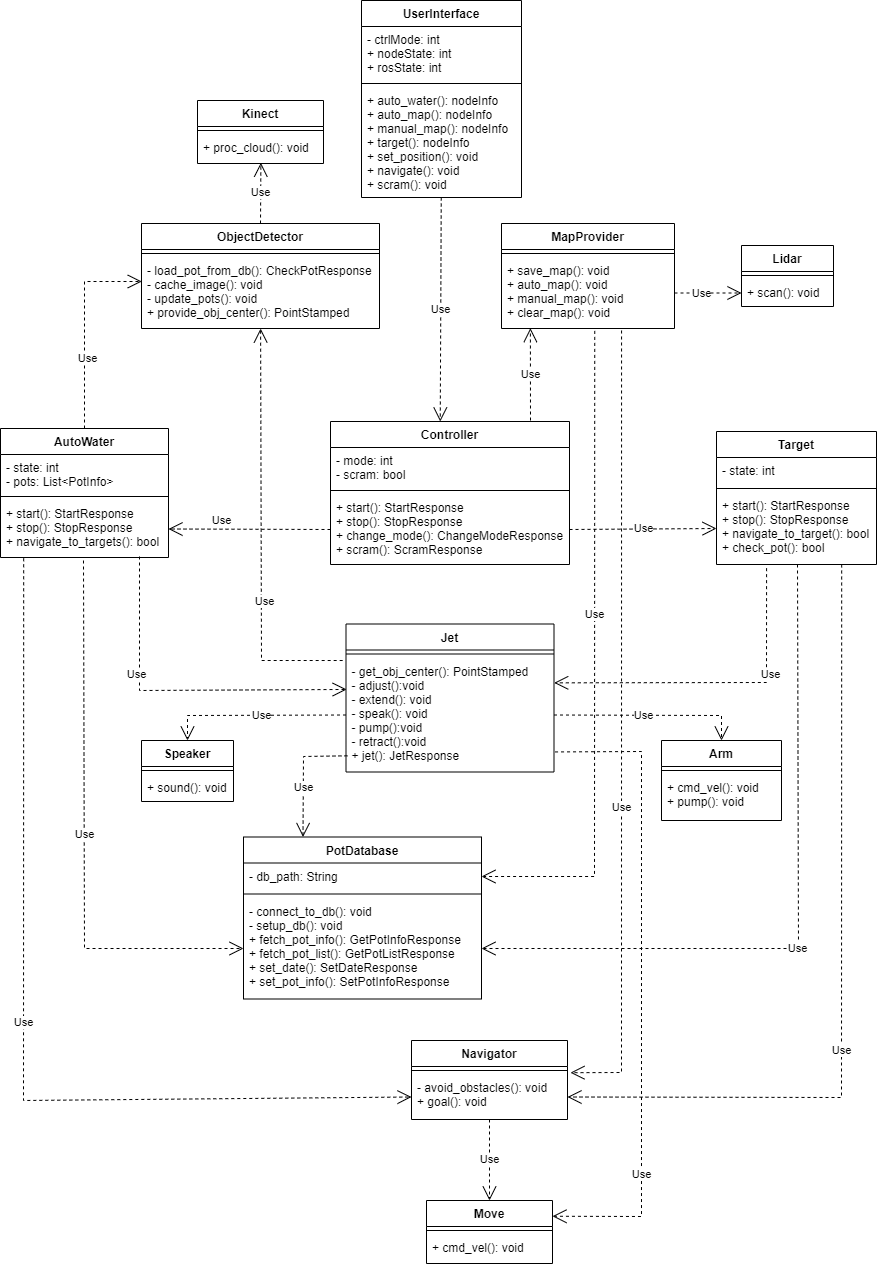
本系统所涉及到的实体主要包含“机器人”、“传感设备”、“浇花机械臂”、“用户交互设备”等，各实体属性以及实体之间的关系如下图所示。



[图8 实体-关系图](file:///D:\Documents\Edge%20Downloads\image\实体-关系图.png)

## 类图

基于对软件需求的分析，我们将整个机器人系统概括为以下14个类，包括主控模块Controller类、任务模块（如AutoWater类）、以及为任务模块提供基础功能的底层模块（如PotDatabase类）等。下面的UML类图展示了系统中各类及其拥有的属性与方法。



[图9 类图](file:///D:\Documents\Edge%20Downloads\image\类图.png)

# 功能需求

## 建图

1. User 通过前端界面向 Controller 发送建图开始消息
2. Controller 向 MapProvider 发送建图开始消息
3. MapProvider 清除原有地图信息, 重新进行建图
4. User 操作机器人移动
5. MapProvider 处理地图信息
6. User 发送建图结束消息, Controller 转发至 MapProvider

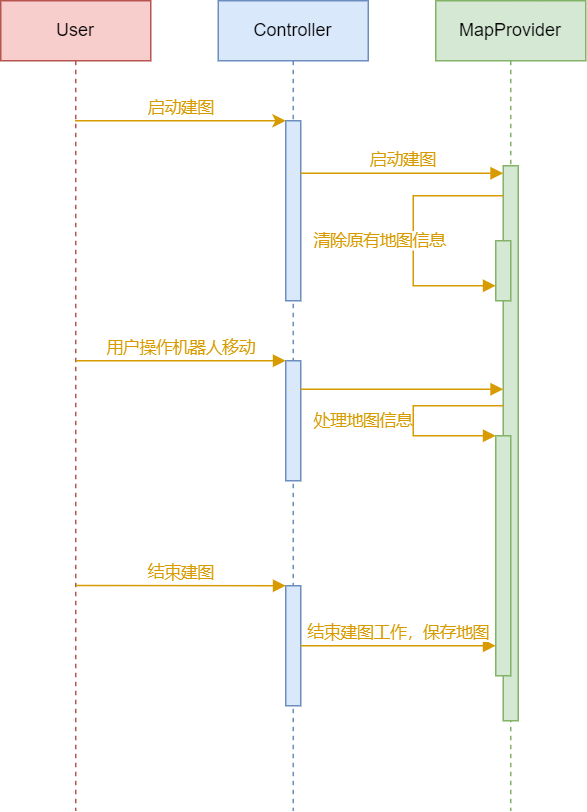


图10 建图时序图

## 地图信息展示

1. User 通过前端界面向 Controller 请求地图信息
2. Controller 转发 MapProvider 提供的地图信息给 User
3. 功能
   1. User 查看某一盆花的详细信息
   2. User 在地图上选定机器人初始位置
   3. User 指示机器人导航到某个特定位置

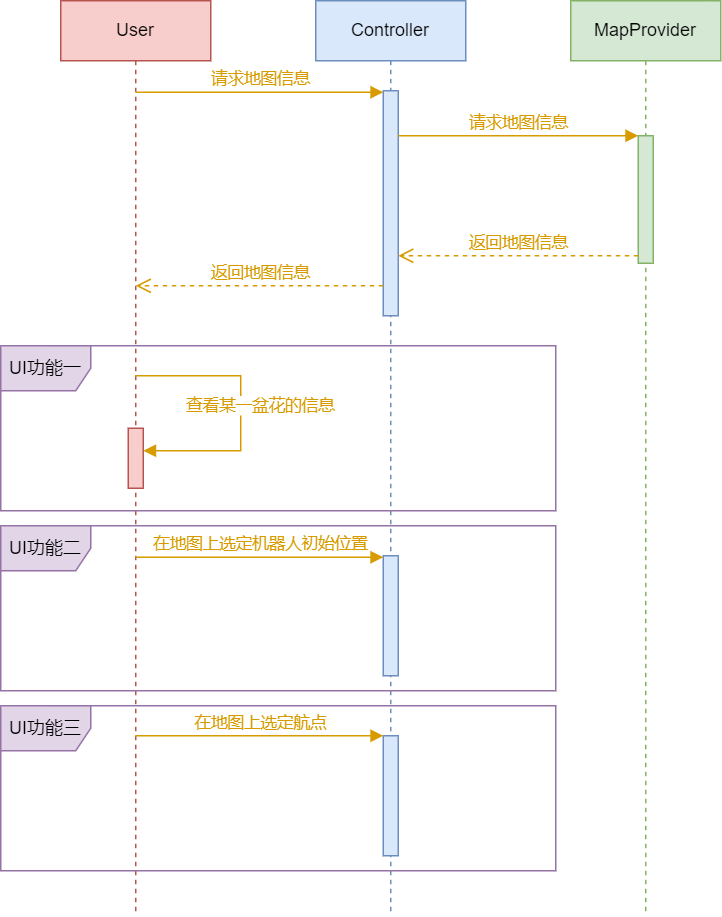


图11 地图信息展示时序图

## 导航与动态避障

1. Controller 向 Navigation 发送目的地
2. Navigation 从 MapProvider 获取地图并导航至目的地
3. 若中途遇到障碍则重新规划路径继续导航
4. Navigation 告知 Controller 已经到达目的地

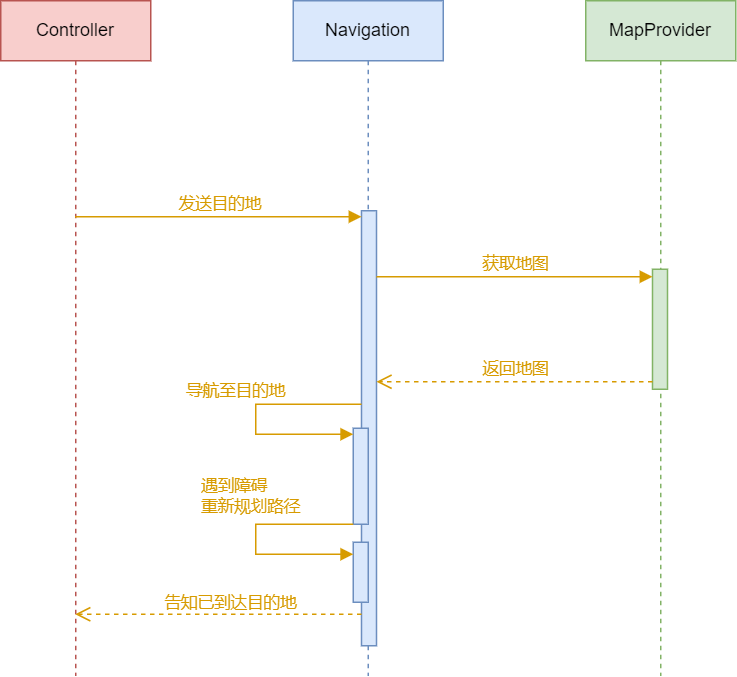


图12 导航时序图

## 花盆识别

1. ObjectDetector 模块识别花盆,并从PotDatabase模块读取已有花盆列表
2. 对比所有已有花盆的位置和检测到的花盆的位置，若位置差距大于一定的阈值则判定为新检测到的花盆
3. ObjectDetector将新花盆信息传递给PotDatabase，PotDatabase写入数据库

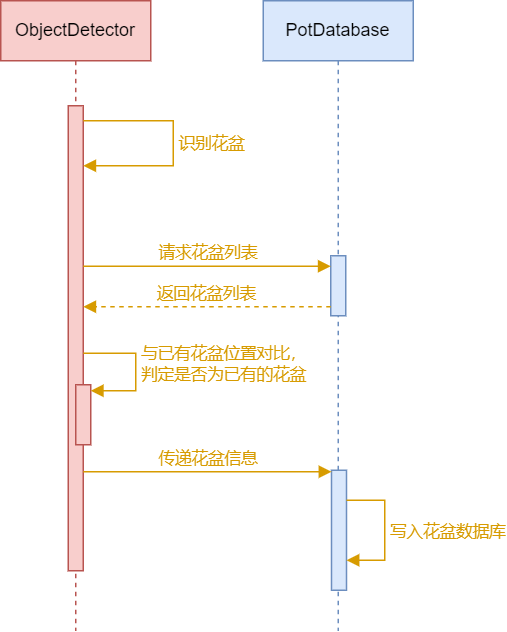


图13 花盆识别时序图

## 浇花

1. Controller 使用功能 3 (导航)移动至指定花盆处, 向 RobotArm 模块发送浇水命令
2. RobotArm 模块使用 ObjectDetector 模块识别花盆并定位浇灌位置
3. RobotArm 模块移动机械臂至指定位置, 启动浇水功能
4. 浇水结束后收回机械臂, 向PotDatabase更新上一次浇水时间，并向 Controller 发送完成信号

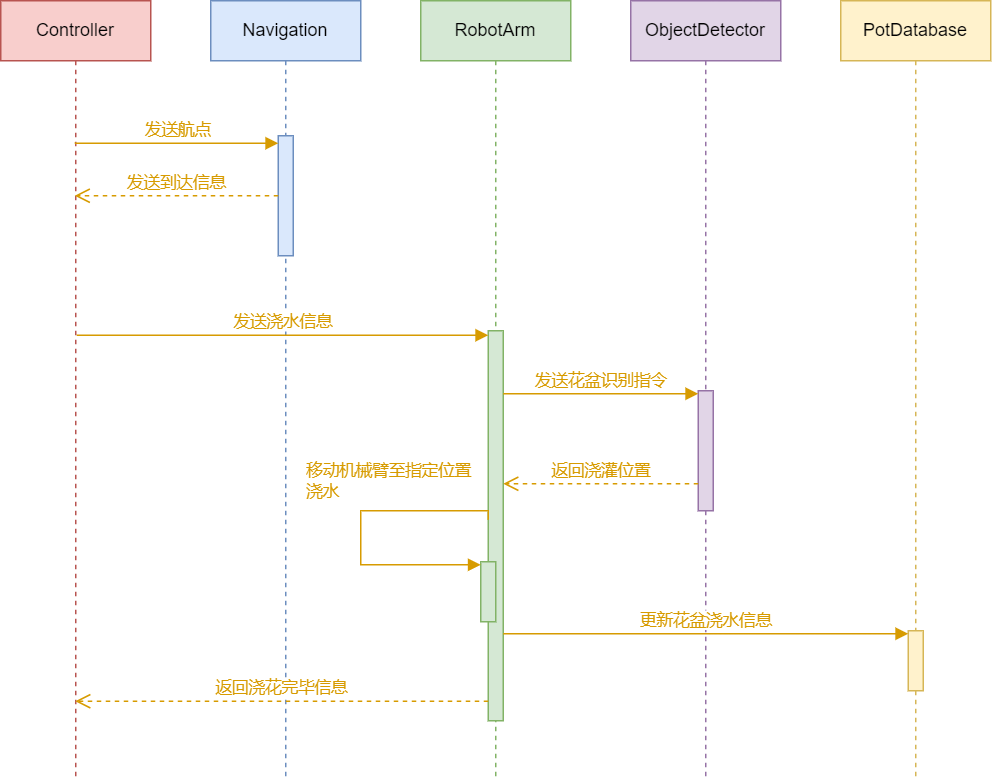


图14 浇花时序图

## 异常处理

### 导航路径出现障碍

1. 在导航路径上出现建图时没有的障碍, 机器人减速并判断障碍是否移动
2. 若障碍移动后不在导航路径上, 机器人继续沿路径移动
3. 若障碍不移动, 机器人尝试重新规划路线以到达目的地
4. 若仍然无法到达目的地, 由 Navigation 模块向 Controller 发送错误消息

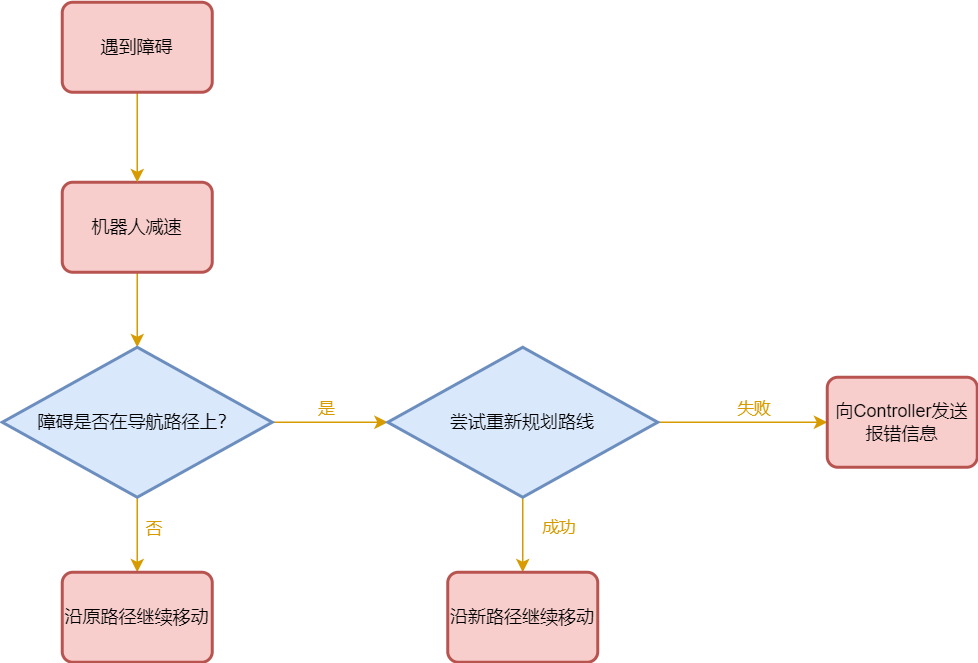


图15障碍处理流程图

### 花盆识别失败

1. 进行定点浇水或自动巡检浇水时没有在指定位置发现花盆
2. 由 ObjectDetector 模块向 Controller 发送错误消息

# 非功能需求

## 性能指标

### 响应时间

针对用户交互面板的控制响应时间要求，系统应确保用户在发出命令或触发操作后，交互界面能够在1秒内作出响应，以确保流畅的用户体验。在极端条件下，即便系统负载较大或存在网络延迟等不利因素，也必须确保响应时间不超过3秒。这涵盖了用户在操作机器人进行建图、巡检、浇水模式切换，以及指定位置浇水和自动浇水过程中的所有命令输入。

对于控制器层面的响应时间要求更为严格，控制器在接收到用户或系统内部组件的指令后，应能在5秒内完成相应的计算、决策及指令下发过程，确保机器人能够迅速启动相应的动作。即使在复杂任务或系统资源紧张的情况下，这一响应时间也不应超过10秒，以维持整个系统的高效运行。

### 功耗

考虑到“启智”机器人硬件本身的能源消耗以及配合使用的笔记本电脑等辅助设备的电量消耗，系统设计应强调节能特性。在非使用时段或长时间无人干预的情况下，机器人与配套设备应具备自动进入低功耗待机模式或完全关闭的功能，最大程度上减少不必要的能耗。

### 异常处理能力

系统在设计时充分考虑了异常处理机制，在执行浇水任务、导航动作等流程中，一旦系统进入异常状态，都应具备较高的自我恢复能力，可以在大多数情况下迅速恢复正常运行，从而保证系统的稳定性和连续性。

## 质量指标

### 系统可用性

本系统拥有高度的可用性，它允许用户通过简洁的触控方式进行全方位的操作控制和互动沟通。此外，系统具有强大的自我检测与恢复机制，大多数常见的异常状况如路径规划故障、震荡时等都能够自动检测并尝试恢复，从而确保系统始终保持在一个较高水平的可用状态。

### 可移植性

本项目基于ROS架构设计，具有出色的可移植性。在仿真阶段，无论是室内模拟环境还是虚拟测试场景，只要平台支持ROS框架，均能顺利迁移和运行。而在实物应用阶段，物理机器人部分则可以在任何集成wpr[1-2]的平台上实现无缝对接，确保不同型号或品牌的机器人只需适当配置，即可实现本项目的各项功能。

### 完整性

本系统的功能设计全面，集成了包括但不限于环境建图、目标物体识别与抓取以及广泛的异常处理机制在内的多项核心功能。这一系列功能紧密结合，形成一个闭环的工作流程，使得机器人系统在温室场景下不仅可以高效完成浇花任务，还能适应复杂的动态环境变化，确保了系统功能的完备性和实用性。

### 效率

本系统在任务执行效率表现上也具有明显优势，机器人可以快速地进行路线规划、花盆识别等核心操作。本系统可以被广泛应用于现代农业和园艺管理中，不仅能显著提升作业效率，还能减少人力资源的依赖，从而降低总体运营成本。此外，该系统具有先进的传感器和缺水识别理论基础，能根据植物的具体生长状况确保精准的养护与管理。

### 健壮性

本系统在健壮性方面表现优异，系统设计时充分考虑了潜在的各种异常情况。在面对这些异常状况时，机器人有能力自我诊断并尝试恢复，从而最大程度地降低系统停摆风险。对于无法立即自我恢复的严重异常，机器人会采取安全待机策略，防止进一步损坏或引发其他问题，直至外部介入进行维护或修复。

### 安全性

本系统在设计时高度重视安全性，首先，机器人在执行物理动作时具备多重安全防护机制，如紧急停止按钮、碰撞检测传感器等，确保在与人或物接触时能及时停止以避免造成伤害。其次，系统内的数据传输与存储均遵守相关的数据加密与安全规范，保护用户隐私及敏感信息不受侵犯。此外，机器人在运行时会严格遵守预设的安全边界和限制区域，确保不会超出预定范围，影响温室或实验室的其他重要设备。

### 维护性

为了简化日常运维工作，本系统提供了丰富的诊断和报告功能，便于管理员监控机器人状态、查找和解决问题。系统各个模块设计成易于拆装和替换的形式，使得硬件维护和软件更新变得简便快捷。同时，系统还支持远程更新和故障诊断，大大降低了现场人工干预的需求。

### 合规性

本系统严格遵守国家和国际的相关法律法规，包括但不限于机器人安全标准、电磁兼容性标准、数据保护法规等。在产品设计、生产和使用过程中，严格遵循绿色制造和可持续发展的原则，确保在整个生命周期内符合环境保护和社会责任的要求。

# 用户界面需求

## 功能特性

### 易用性与直观性

用户界面设计应简洁明了，图标和文字表述清晰，确保新手能快速理解和上手。界面布局逻辑性强，功能分类明确，提供简明的引导教程和操作提示，以帮助实验室研究人员和花卉种植管理人员等使用者高效轻松使用机器人系统。

### 个性化定制

用户可以根据使用场景中的盆栽浇水需求设定不同的浇水模式以满足个性化的温室盆栽浇水需求。同时，用户界面应支持自定义巡检路线的创建与保存，以及在必要时调整机器人行走速度、避障距离等参数。

### 异常提醒与处理

当机器人遇到建图失败、花盆识别错误、路径规划受阻等问题时，用户界面应及时显示警告信息，并指导用户如何应对。例如，当机器人检测到花盆识别结果与预计不符时，会在前端提醒用户异常情况。

### 执行记录

系统应保存机器人执行记录日志文件，以供使用者、维护人员进行查看以确保机器人处于正常工作的模式，同时为维护人员提供参考信息。

## 可视化呈现

### 地图展示

用户界面应具备三维或二维地图视图，实时展示温室内部布局，包括花盆位置、机器人的当前位置及行动轨迹。

### 状态反馈

机器人执行任务过程中的实时反馈至关重要，如浇水状态、花盆缺水与否、巡检结果等信息应在界面上以直观形式展现。

### 外观设计

用户界面应采用舒适、柔和的色调，减轻长时间操作带来的视觉疲劳，并结合品牌形象进行专业设计，确保界面美观大方，符合人体工学原理，增强用户的操作体验。

## 交互方式

### 触控界面

系统应配备触屏式的人机交互界面，用户可以直接触摸屏幕来指定机器人初始位置、选择浇花位置、编辑巡检路线等，并能实时查看地图信息、花盆状态以及任务进度。

# 运行与开发环境

## 运行环境

1. 启智ROS配套开发电脑
2. 启智ROS机器人
3. 启智ROS机器人配套机械臂
4. 单片机、继电器、杜邦线等配套材料
5. 直流水泵、软胶管

## 软件环境

1. Ubuntu 18.04 LTS
2. ROS Melodic
3. Rviz可视化平台
4. Gazebo仿真环境
5. 启智ROS机器人的模块与驱动代码