**浇花机器人——** **温室管家**

**软件设计说明书**

**SDD-03**

**V3.1**

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 |
| 1 | Le | 主编、小组讨论、数据库设计、内部接口、浇水、运行与开发环境 |
| 2 | Hao7un | 审核、小组讨论、技术体系结构、支撑体系结构、花盆识别 |
| 9 | LTT | 小组讨论、系统用户界面、外部接口 |
| 6 | Kun | 小组讨论、软硬件体系结构、导航与动态避障 |
| 5 | Mxode | 小组讨论、建立地图 |

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 | 2024.4.23 | Le | Hao7un | 软件设计说明初稿 |
| V1.1 | 2024.5.11 | Le | Hao7un | 完善了图题和表题 |
| V1.2 | 2024.5.12 | Le | Hao7un | 修改了控制器各节点状态转移图 |
| V1.3 | 2024.5.14 | Kun | Le | 更新了软硬件体系结构图和需求 |
| V1.4 | 2024.5.20 | Le | Hao7un | 更新了数据库表和类图 |
| V1.5 | 2024.5.20 | Le | Hao7un | 修改了缩略词和正文的部分表述 |
| V1.6 | 2024.5.21 | Le | LTT | 更新了系统用户界面设计图 |
| V2.0 | 2024.5.22 | Le | Kun | 更新了所有inspection节点信息 |
| V2.1 | 2024.5.23 | Le | LTT | 调整了参考引用文献 |
| V2.2 | 2024.5.24 | Le | LTT | 修改了浇花节点arm时序图 |
| V3.0 | 2024.5.26 | Le | Hao7un | 将点云全部更新为YOLO识别 |
| V3.1 | 2024.6.10 | Le | LTT | 修改了部分需求描述和缩略词 |

# 1 范围

## 项目概述

### 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，集成自动建图、动态避障、语音交互等技术，通过实现自动浇灌功能，降低人工成本并提高温室管理的智能化水平，适用于对番茄等温室作物的精细化照料。

### 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，具体而言包括以下功能。

* 6种工作模式，对应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。
* 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。

### 非功能性需求

* 可靠性与稳定性：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，能够实时反馈任务进度，并在异常情况发生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。

### 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

本软件设计说明的主要用途是阐述软件的设计和实现规范，以确保所有开发人员和相关利益相关者对软件的结构和功能有清晰的理解。本软件设计说明文档为软件开发设计提供了方案细节并且确保了系统具体实现的一致性。

### 内容组织

* 项目任务概要
* 需求概述
* 数据库设计
* 体系结构设计
* 接口设计
* 详细设计
* 运行与开发环境
* 需求可追踪说明

## 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（Robot Operating System）

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/全称对应表

| **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 |
| SLAM | Simultaneous Localization and Mapping |
| YOLO | You Only Look Once |

## 引用文档

[1] Nakanwagi M J, Sseremba G, Kabod N P, et al. Identification of growth stage-specific watering thresholds for drought screening in Solanum aethiopicum Shum[J]. Scientific Reports, 2020, 10(1): 862.

# 需求概述

## 业务需求

本机器人需满足温室种植业的核心业务需求，即高效精准的灌溉作业，具体表现在以下几点。

* 多模式灌溉：机器人具备三种可调节的工作模式以满足不同的灌溉需求。**巡检维护模式**下，机器人按预设路线检查所有花盆位置，维护花盆真实位置信息与数据库花盆信息的一致性。**定点浇水模式**下，机器人可根据用户制定的花盆位置进行顶点浇水。**自动巡检浇水模式**下，机器人在巡检过程中自动检测并为缺水花盆浇水。
* 环境感知与动态避障：机器人可利用雷达和摄像头等传感设备实时识别周围环境，并在执行任务的过程中灵活调整路径、安全避开障碍物，确保浇灌任务顺利完成。
* 人机交互：机器人提供用户友好的人机交互界面，支持一键建图、标注航点、切换任务、查询植物状态等功能，增强用户对机器人作业的控制力和监控能力。

## 数据需求

数据需求主要包括以下几个方面。

* 场景地图数据：保存温室场景的地图数据，指导机器人规划任务路线。
* 花盆位置信息：存储每个花盆在温室内的位置坐标，用于在机器人巡航时定位目标花盆。
* 水分监测数据：记录每个花盆的实时水分状况，作为是否需要浇水以及量化灌溉量的根据。
* 历史任务数据：保存用户设定的工作模式和特定的历史浇水请求，便于用户查看任务与工作日志。
* 机器人状态数据：包括机器人的工作模式、运动参数、连接情况和故障信息，用于监控机器人的工作状态和性能。

## 功能性需求

* 自主建图：机器人能够利用主动SLAM算法自主构建温室内部地图，为导航和任务规划提供数据基础。
* 导航与动态避障：定点导航，且在导航的过程中实现动态避障。
* 花盆识别：采集花盆图片、三维坐标等信息，用于更新数据库中花盆信息。
* 浇花：机械臂抬升软管对准花盆，控制水泵通电浇水，并更新数据库中的花盆浇水信息。
* 数据库服务：管理植物在温室中的位置，含水量等数据和历史任务数据等。
* 实时监测：用户可实时监测花盆水分状态，确保及时发现并解决植物缺水的问题。
* 语音提示：支持播放语音进度提示和语音警示词，可提升用户体验，并保证作业周围环境安全。

## 非功能性需求

* 可靠性与稳定性：机器人应能够长时间连续工作，并在复杂的温室场景下保持稳定运行。
* 异常处理：遇到异常时能够自动采取恢复措施或及时向用户反馈问题，确保系统持续可用。

# 数据库设计

## 植物数据表

植物数据表记录了每株植物的id、名称、地图中位置，以及缺水情况。下面是植物数据表对应的数据字典。

表2 植物数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 约束 | 说明 |
| plant\_id | INT | 是 | UNIQUE | 植物编号 |
| pot\_pose | BLOB | 否 | NOT NULL | 植物位置 |
| robot\_pose | BLOB | 否 | NOT NULL | 机器人浇水位置 |
| last\_water\_date | DATETIME | 否 | NOT NULL | 上次浇水的日期 |
| active | BOOLEAN | 否 | NOT NULL | 自动巡检是否浇水 |
| data | BLOB | 否 | NOT NULL | 花盆点云数据 |
| picture | BLOB | 否 | NOT NULL | 花盆图片数据 |

## 任务数据表

为了便于用户查看历史浇灌任务记录，数据库中可存放用户历史指令记录及其对应的航点信息。每条数据包括任务id、任务地点、任务时间以及工作模式字段。下面是指令数据表对应的数据字典。

表3 历史任务数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 约束 | 说明 |
| task\_id | INT | 是 | UNIQUE | 指令Id |
| task\_date | DATETIME | 否 | NOT NULL | 任务时间 |
| task\_mode | VARCHAR (10) | 否 | NOT NULL | 任务工作模式 |
| task\_status | BOOLEAN | 否 | NOT NULL | 任务完成情况 |

# 体系结构设计

## 软硬件体系结构

系统的层次自顶向下分别为：用户层、控制层、服务层、功能层、硬件层。

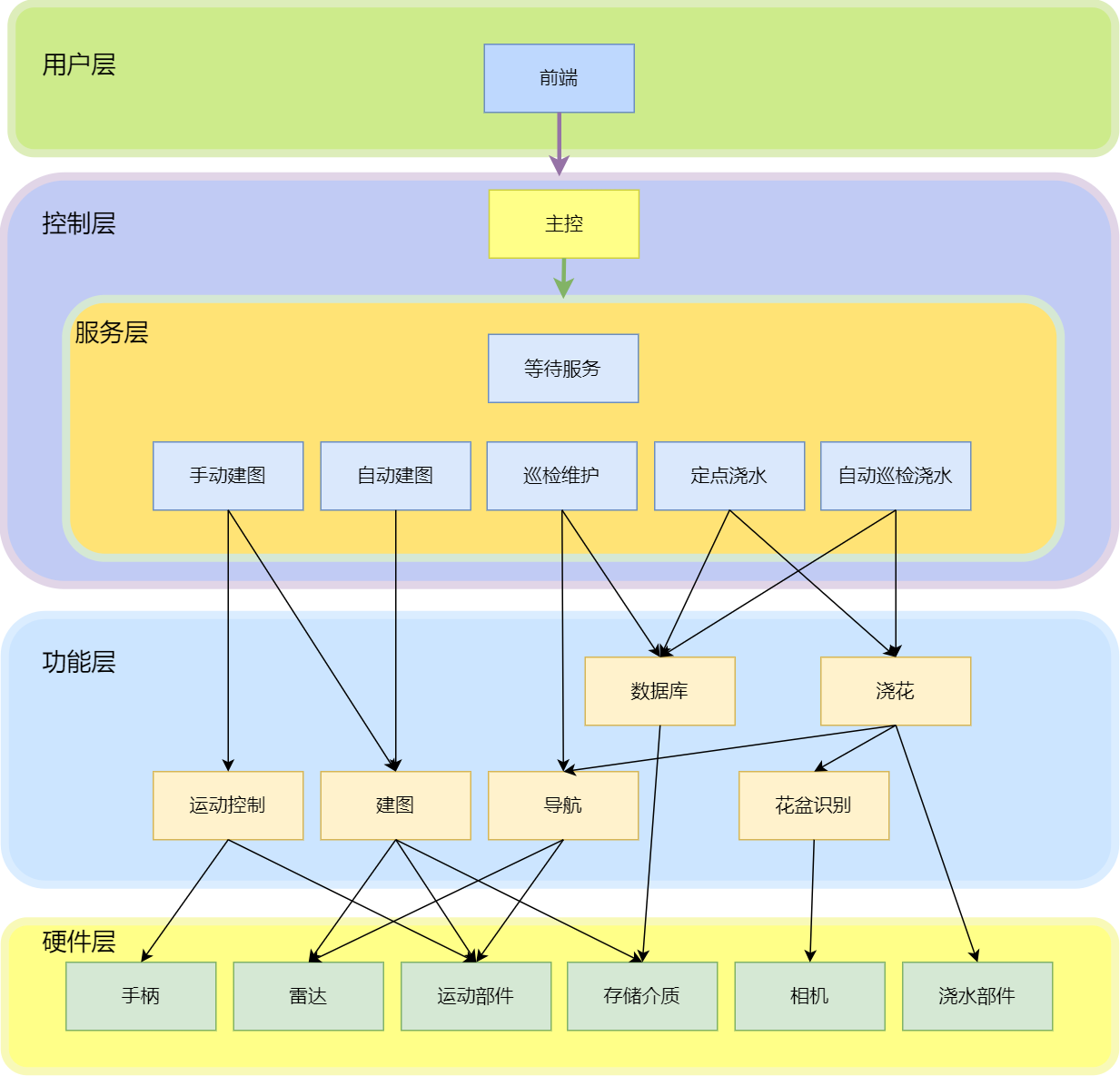


图1 体系结构层次图

以下是本项目体系结构的设计类图：

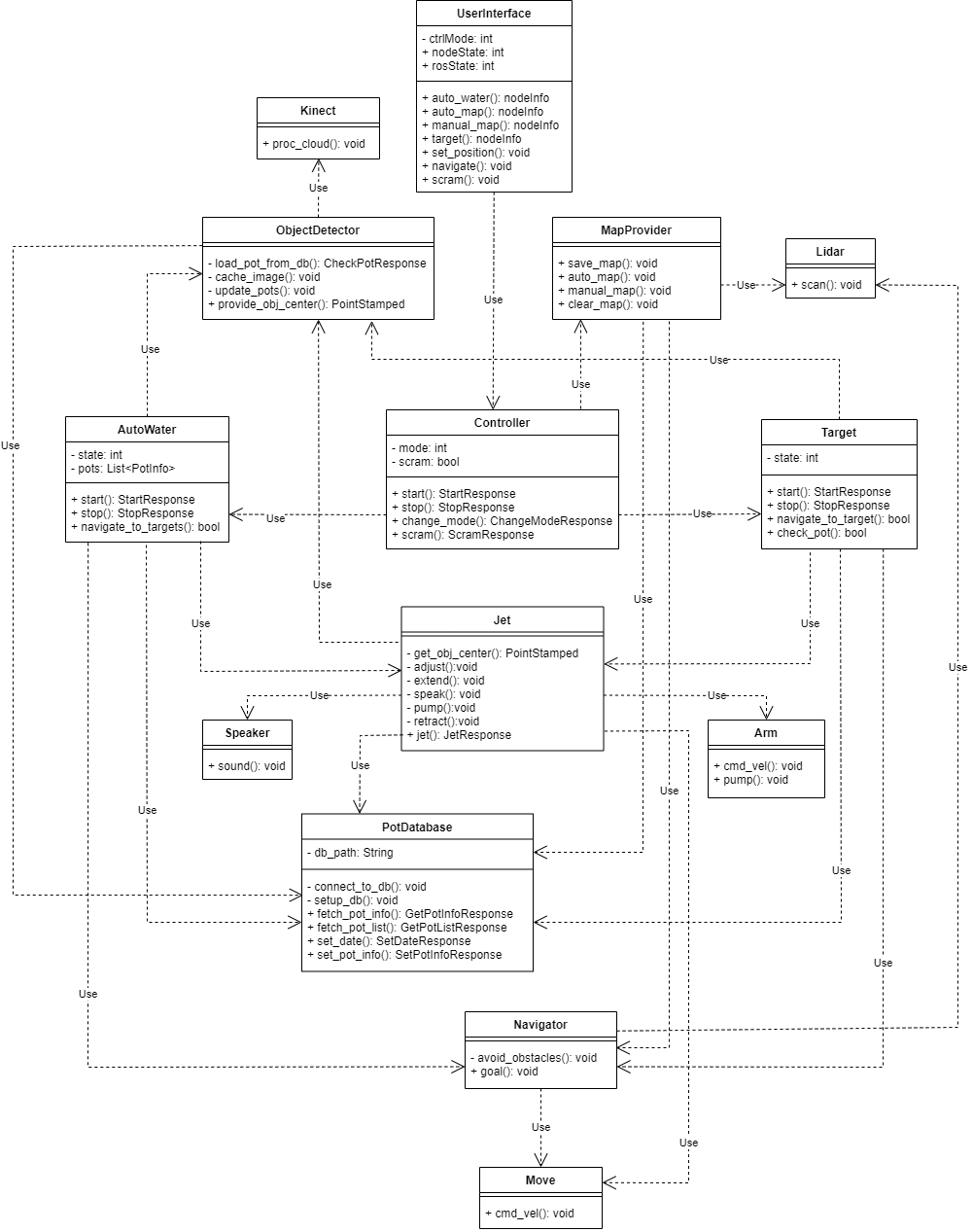


图2 体系结构类图

### 用户层

用户层负责传送用户指令，并将ROS返回的多种数据进行可视化。用户通过该层调用建图和浇水等服务，并可通过用户界面实时查看机器人的工作进度。在这一过程中，用户层将指令通过ROS Bridge传递给ROS，并得到ROS返回的地图信息、浇花数据、工作进度，机器人状态等信息。

### 控制层

控制层负责服务进程的调度，确保主控在同一时间只允许机器人执行一种服务。高优先级的指令（如急停）会中止正在进行的低优先级服务（如浇花，建图等）。该层将用户指令分发给下方服务层的5个节点。

### 服务层

服务层负责实现5种用例。每一种用例由一个单独的服务节点实现，同一时间只有一个服务节点被启动。服务层节点通过调用功能层节点实现建图，浇花等用例。

### 功能层

功能层负责实现相对独立且特定的功能，如建图、导航、浇花、花盆识别等。数据库功能通过SQLite实现，其余功能通过ROS官方包或者官方包的封装实现。

### 硬件层

该层包括机器人系统的各种硬件设备，主要包括三类：第一类是与功能层双向交互的存储介质；第二类是输入设备，包括手柄、雷达、相机；第三类是输出设备，包括运动部件、浇水部件。

## 技术体系结构

本项目的技术体系结构如下图所示：

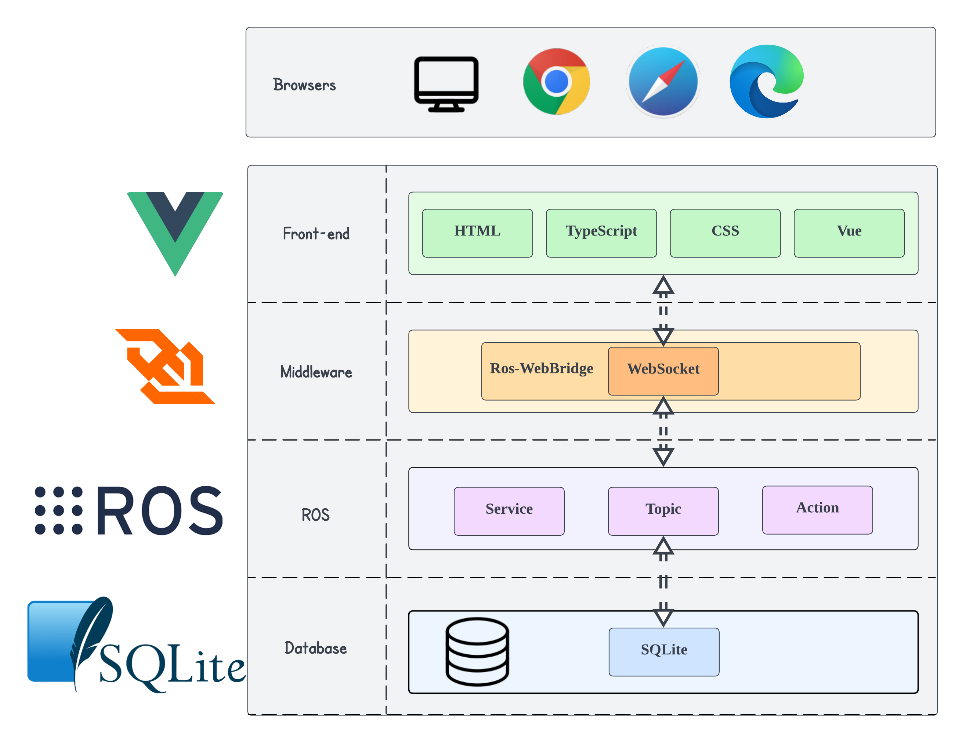


图3 技术体系结构层次图

### 前端界面技术

在本项目中，前端用户界面采用Vue框架进行搭建。Vue是一款轻量级、高效且灵活的前端框架。本项目主要利用其响应式数据绑定和组合式API来构建一个动态且响应迅速的浇水机器人用户界面。此外，本项目还使用Vuetify组件库来加速UI的开发，该库提供了丰富的预制组件，遵循Material Design规范，能够保证用户界面的美观性和功能性。

### 通信技术

本项目采用ROS-WebBridge技术来进行前端用户界面与ROS层之前的信息交互，其包含一个WebSocket 服务器ros\_bridge server，允许通过网络进行实时通信。客户端可以通过WebSocket连接到这个服务器，并使用JSON格式的消息与 ROS 交互。ROS-WebBridge提供的服务包括发布和订阅 ROS 主题，请求和响应服务等。

### 机器人技术

本项目主要基于ROS技术对机器人项目进行开发，ROS是适用于机器人的开源元操作系统。它提供了操作系统应有的服务，为开发者提供了硬件抽象、底层设备控制、进程间通讯等服务。此外，ROS还集成了可视化工具，如RViz和Gazebo，用于调试并模拟机器人与环境的交互行为。

### 数据库存储技术

数据存储主要使用嵌入式开发中常用的SQLite数据库。SQLite提供了一种轻量级的磁盘基存储解决方案，不需要单独的服务器进程，可以轻松集成到各种应用中。相比其他数据库系统如MySQL或Oracle，SQLite占用更少的资源和内存，且存储文件更小，非常适用于嵌入式系统开发。

## 支撑体系结构

### 部署环境

* 软件环境
  + 操作系统: Ubuntu 18.04 LTS是一个稳定的Linux发行版，广泛用于各种服务器和桌面应用，提供必要的库和支持，特别适合开发和部署ROS应用。
  + ROS版本: ROS Melodic是专为Ubuntu 18.04 LTS定制的ROS版本，支持C++11，提供了改进的移动库和新的功能，如兼容Python 3。
  + 仿真环境依赖：仿真环境的主要依赖为Rviz和Gazebo9，用于在开发、测试环节中对机器人进行仿真模拟与测试。
* 硬件环境
  + 处理器: 至少需要四核处理器，例如Intel i5或更高性能的处理器，以处理图像识别和传感器数据分析等计算密集型任务。
  + 内存: 至少8GB RAM，以便于处理大量的实时数据和执行多任务操作。
  + 存储: 至少256GB的硬盘空间，用于存储日志文件、系统更新以及用户数据。
  + 网络连接: 支持无线或有线网络，确保机器人能够接收远程命令和反馈数据到前端用户界面。

### 部署策略

* 初始部署
  + 安装操作系统和依赖项: 首先在机器人的计算模块上安装Ubuntu 18.04，并安装所有必需的软件依赖项，包括ROS Melodic和其他必需的编程库和工具(如代码编辑器VS Code等)。
  + 配置网络: 配置机器人的网络设置，确保其可以通过本地网络或互联网与用户前端通信。
  + 部署应用程序: 通过ROS的包管理系统部署浇花机器人的所有ROS包和节点，确保所有驱动和功能包正确加载并正常运作。
* 维护和更新
  + 定期更新: 定期检查和安装操作系统和ROS的更新程序，从浇花机器人代码仓库中获取更新。
  + 监控和日志记录: 对浇花机器人运作中间过程进行日志记录策略，以跟踪机器人的性能及任何潜在的问题。

# 接口设计

## 系统用户界面

### 总体设计

用户界面由顶栏, 花盆信息侧栏, 地图显示界面, 模式相关操作侧栏和底栏构成。

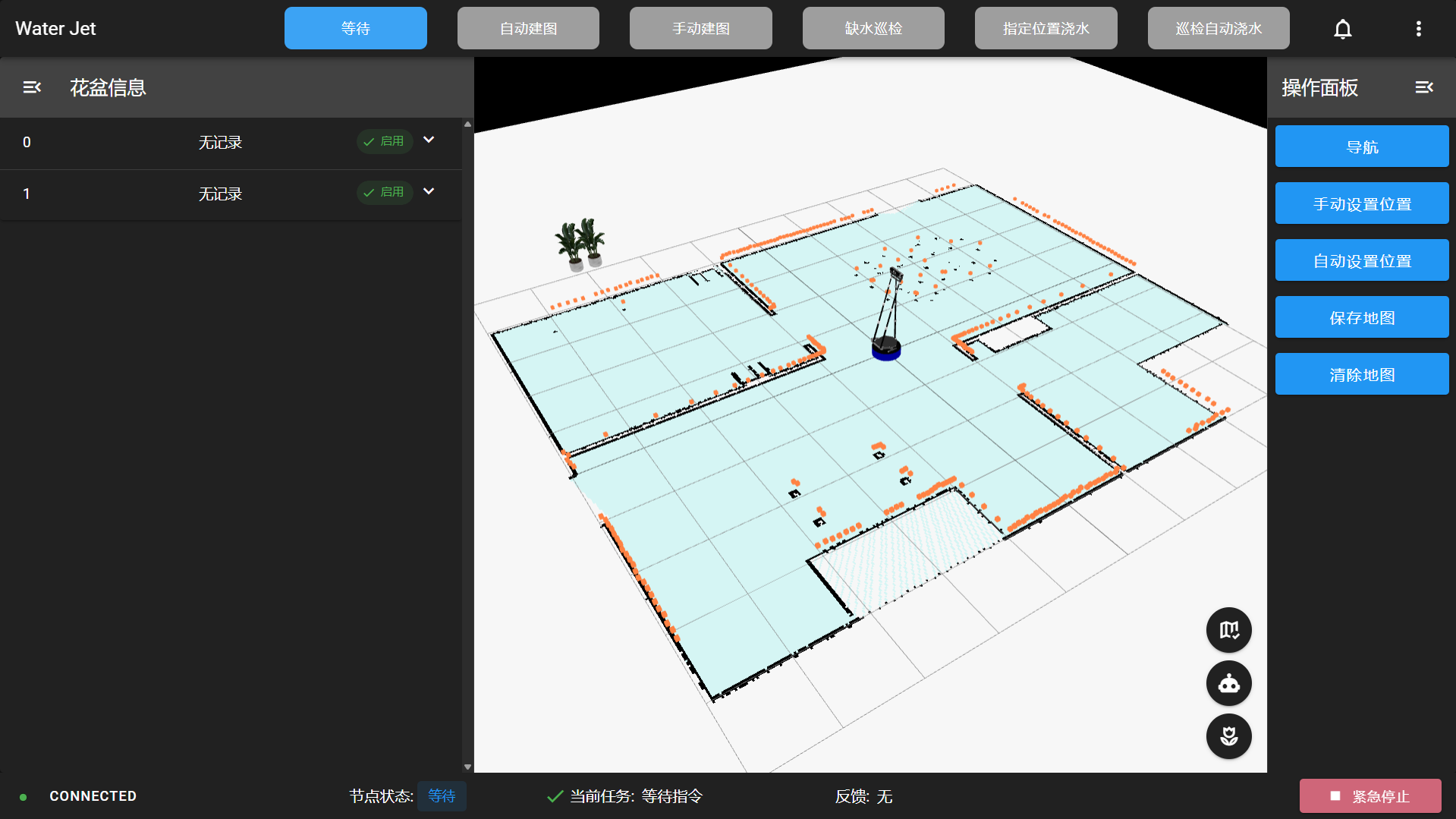


图4 总体用户界面设计图

* 顶栏
  + 项目名Water Jet
  + 模式切换按键组
  + 通知和相关信息按钮
* 花盆信息侧栏
  + 显示花盆信息
  + 可以折叠
  + 可以与地图中的花盆位置对照
* 地图显示界面
  + 3D 地图显示
  + 地图显示相关选项
* 模式相关操作侧栏
  + 针对每个模式的操作选项
* 底栏
  + ROS 连接状态, 可以点击以重新连接到 ROS
  + 当前工作状态, 显示当前模式的工作状态
  + 急停按钮, 按下后机器人的运动会立即停止

### 等待界面

等待界面用户可以对查看地图信息, 花盆信息, 进行地图相关操作。

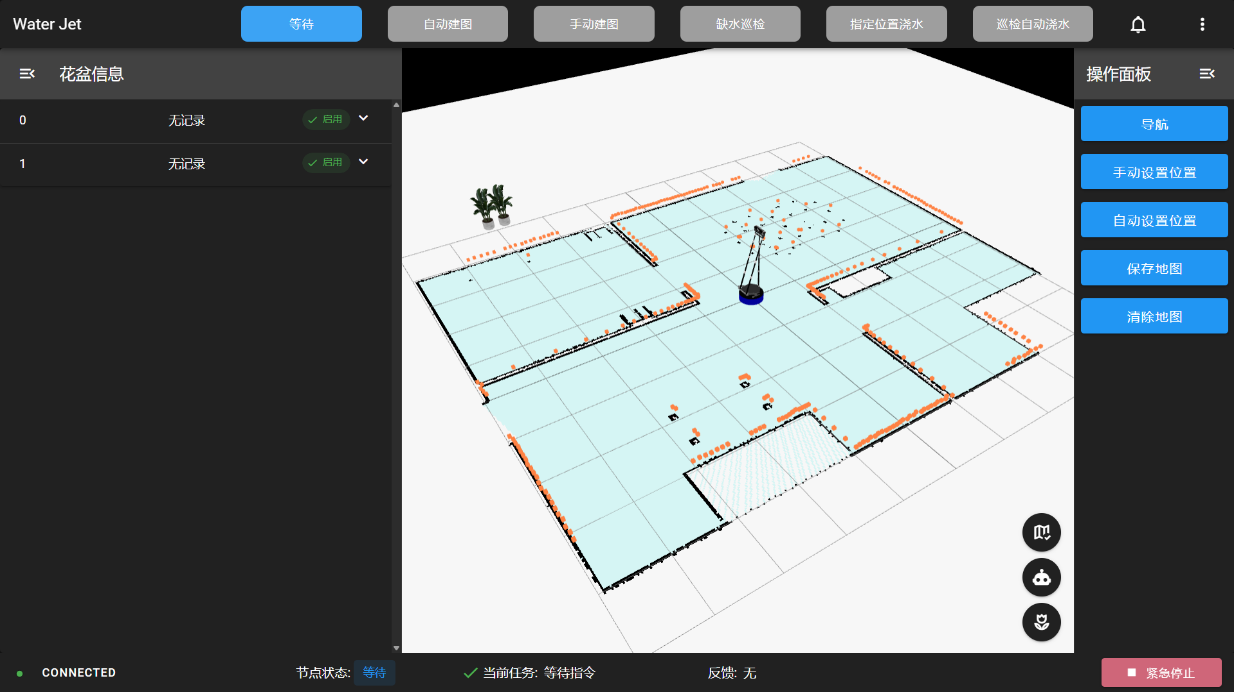


图5 等待界面设计图

### 自动建图界面

进入自动建图界面后, 机器人会自动开始建图, 此时地图的相关选项将不可用, 以避免冲突。花盆信息将随建图自动更新。

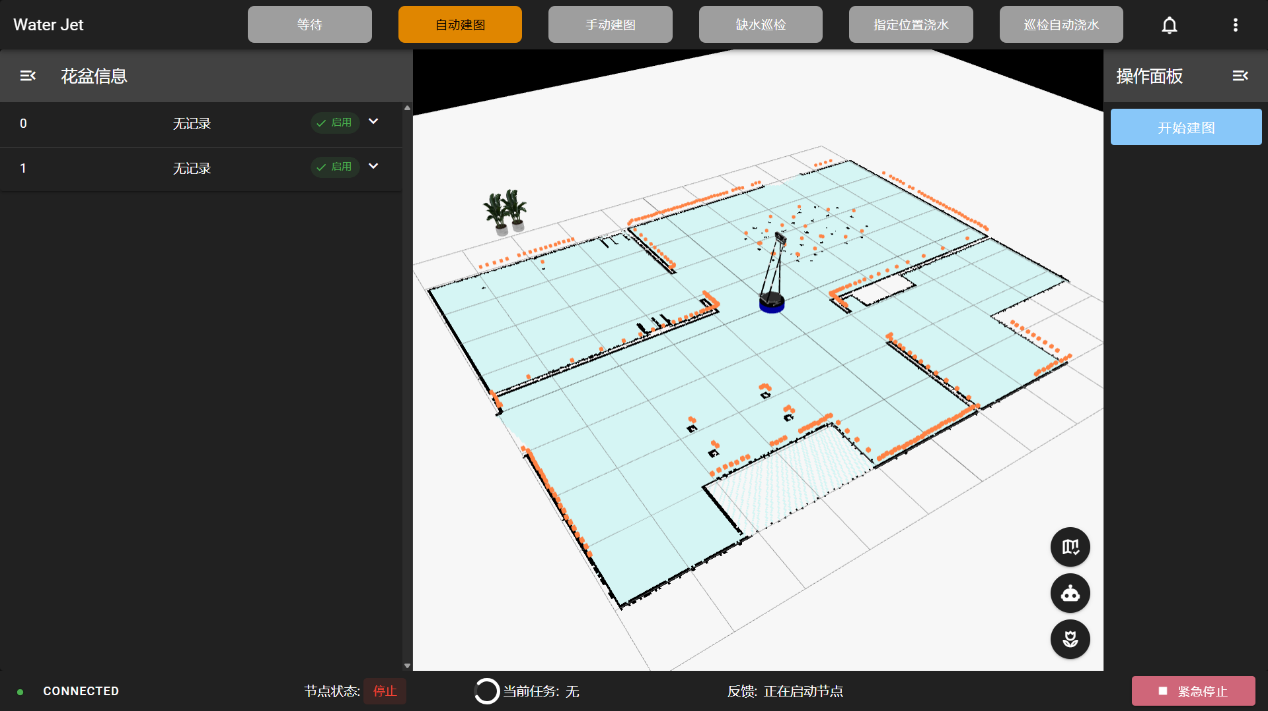


图6 自动建图界面设计图

### 手动建图界面

进入手动建图界面后, 机器人会根据用户操作开始建图, 此时地图的相关选项将不可用, 以避免冲突。花盆信息将随建图自动更新。

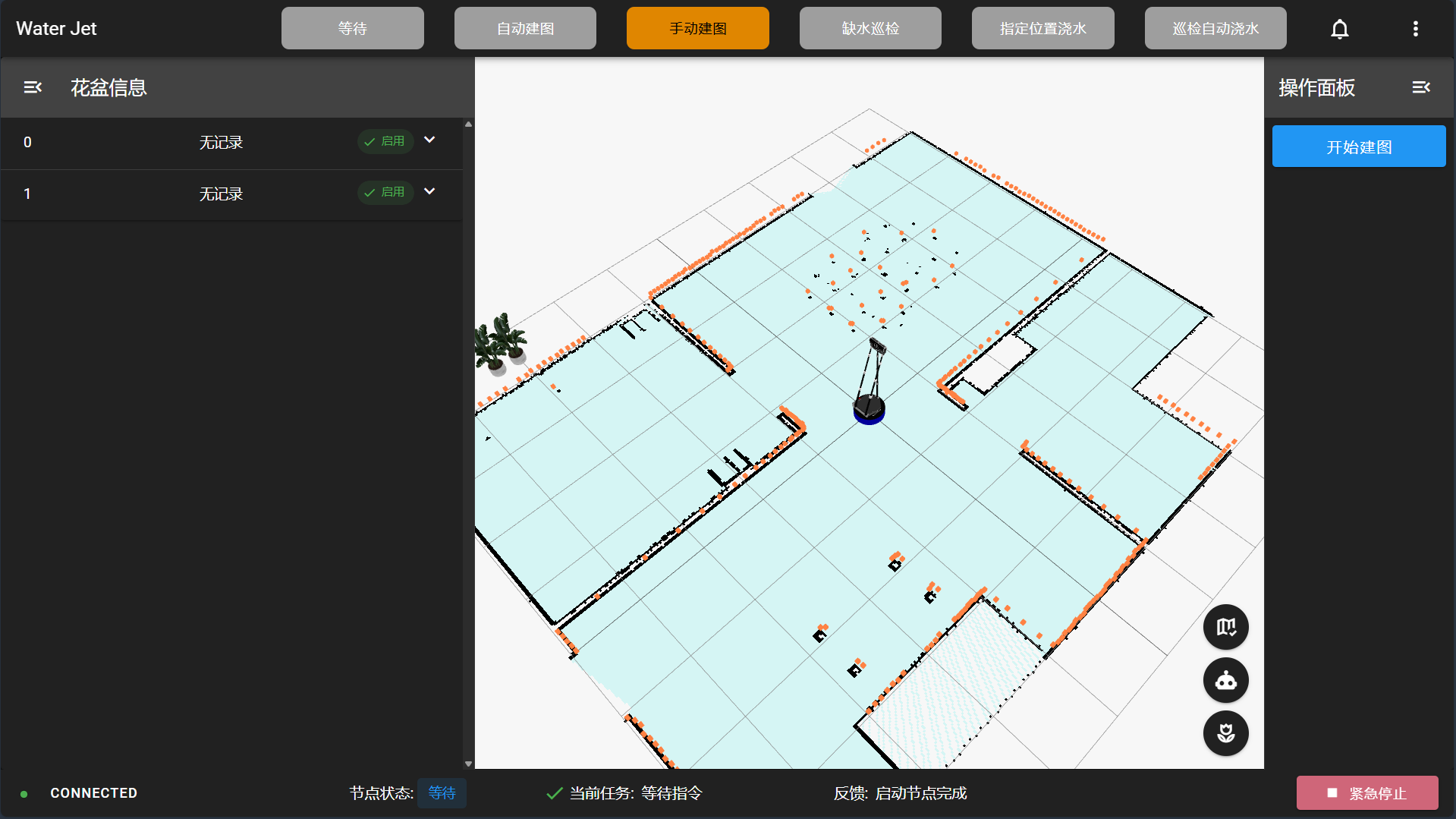


图7 手动建图界面设计图

### 缺水巡检界面

进入缺水巡检界面后, 机器人将根据花盆信息开始规划路径并巡检。地图相关选项将不可用, 以避免冲突。花盆信息栏将显示当前目标花盆信息。

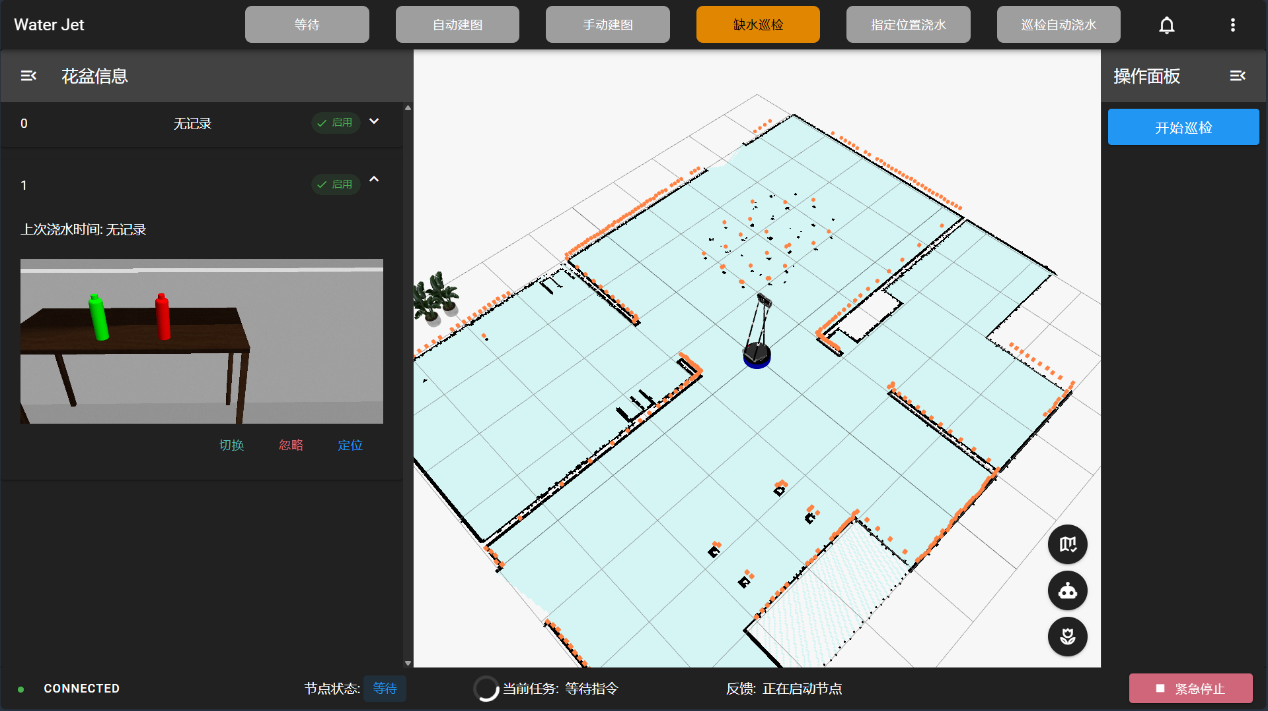


图8 缺水巡检界面设计图

### 指定位置浇水界面

进入指定位置浇水界面后, 用户需要点击记录花盆队列, 选择需要浇水的花盆, 随后点击开始, 机器人将对指定花盆进行浇水。地图相关选项将不可用, 以避免冲突。花盆信息栏将显示当前目标花盆信息。

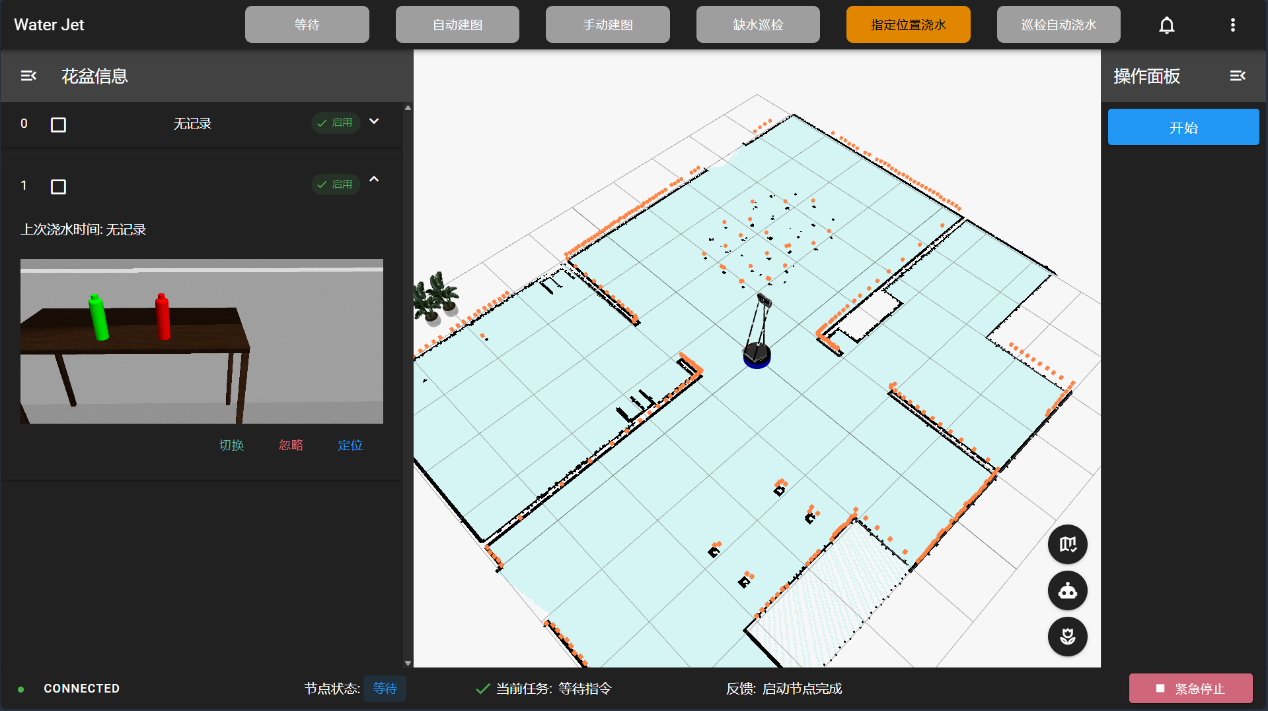


图9 指定位置浇水界面设计图

### 巡检自动浇水界面

进入巡检自动浇水界面后, 机器人将自动巡检并对花盆浇水。地图相关选项将不可用, 以避免冲突。花盆信息栏将显示当前目标花盆信息。

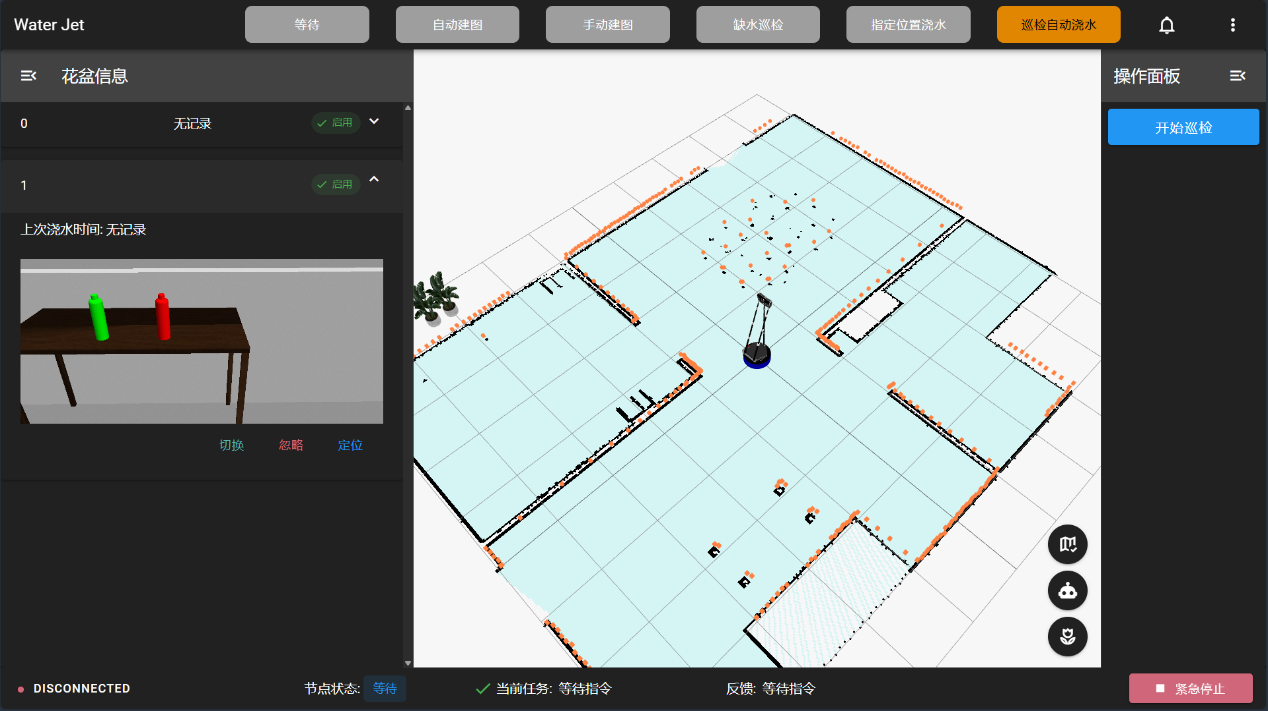


图10 巡检自动浇水界面设计图

## 控制器接口设计

### Subscribed Topics

表4 core订阅话题表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 作用 | 订阅者 |
| \_cmd\_vel | /geometry\_msgs/Twist.msg | 控制机器人运动 | core |

### Published Topics

表5 core发布话题表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 订阅者 |
| core | info | Info.msg | 发布全局状态信息 | frontend |
| core | /cmd\_vel | /geometry\_msgs/Twist.msg | 控制机器人运动 |  |

* 节点省略前缀 /ctrl/
* 名称省略前缀 /ctrl/
* 类型省略前缀 /controller/msg/

### Services

表6 core Services通信表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 调用者 |
| core | start | Start.srv | 开始 | frontend |
| core | stop | Stop.srv | 停止 | frontend |
| core | change\_mode | ChangeMode.srv | 更改模式 | frontend |
| core | scram | Scram.srv | 切换急停 | frontend |
| pending | pending/start | Start.srv | 启动  pending  节点功能 | core |
| pending | pending/stop | Stop.srv | 停止  pending  节点功能 | core |
| pending | pending/save\_map | SaveMap.srv | 保存地图 | frontend |
| pending | pending/clear\_map | ClearMap.srv | 清除已保存地图 | frontend |
| pending | pending/auto\_init\_pos | AutoInitPos.srv | 自动设置初始位置 | frontend, core |
| pending | pending/manual\_init\_pos | ManualInitPos.srv | 手动设置初始位置 | frontend |
| auto\_map | auto\_map/start | Start.srv | 启动 auto\_map  节点功能 | core |
| auto\_map | auto\_map/stop | Stop.srv | 停止 auto\_map  节点功能 | core |
| manual\_map | manual\_map/start | Start.srv | 启动 manual\_map  节点功能 | core |
| manual\_map | manual\_map/stop | Stop.srv | 停止 manual\_map 节点功能 | core |
| inspection | inspection/start | Start.srv | 启动inspection  节点功能 | core |
| inspection | inspection/stop | Stop.srv | 停止inspection  节点功能 | core |
| target | target/start | Start.srv | 启动 target  节点功能 | core |
| target | target/stop | Stop.srv | 停止 target  节点功能 | core |
| auto\_water | auto\_water/start | Start.srv | 启动 auto\_water 节点功能 | core |
| auto\_water | auto\_water/stop | Stop.srv | 停止 auto\_water 节点功能 | core |

* 节点省略前缀 /ctrl/
* 名称省略前缀 /ctrl/
* 类型省略前缀 /controller/srv/

### Actions

表6 core Actions通信表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 调用者 |
| auto\_map | auto\_map/auto\_init\_map/goal | InitMap.action | 开始自动建图 | frontend |
| auto\_map/auto\_init\_map/cancel | 取消建图 |
| auto\_map/auto\_init\_map/feedback | 反馈信息 |
| auto\_map/auto\_init\_map/status | 当前goal 状态 |
| auto\_map/auto\_init\_map/result | 结果 |
| manual\_map | auto\_map/auto\_init\_map/goal | InitMap.action | 开始手动建图 | frontend |
| auto\_map/auto\_init\_map/cancel | 取消建图 |
| auto\_map/auto\_init\_map/feedback | 反馈信息 |
| auto\_map/auto\_init\_map/status | 当前goal 状态 |
| auto\_map/auto\_init\_map/result | 结果 |
| inspection | inspection/inpect/goal | Inspect.action | 开始巡检 | frontend |
| inspection/inpect/cancel | 取消操作 |
| inspection/inpect/feedback | 反馈信息 |
| inspection/inpect/status | 当前 goal 状态 |
| inspection/inpect/result | 结果 |
| target | target/set\_target/goal | SetTarget.action | 设置浇水花盆 | frontend |
| target/set\_target/cancel | 取消指定目标 |
| target/set\_target/feedback | 反馈信息 |
| target/set\_target/status | 剩余 goal 信息 |
| target/set\_target/result | 结果 |
| auto\_water | auto\_water/water/goal | Water.action | 开始巡检浇水 | frontend |
| auto\_water/water/cancel | 取消操作 |
| auto\_water/water/feedback | 反馈信息 |
| auto\_water/water/status | 当前 goal 状态 |
| auto\_water/water/result | 结果 |

* 节点省略前缀 /ctrl/
* 名称省略前缀 /ctrl/
* 类型省略前缀 /controller/action/

## 内部接口设计

### database

#### Services

表7 database Services通信表

|  |  |  |  |
| --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 |
| database | /pot/list | PotList.srv | 所有花盆详细信息 |
| database | /pot/set | SetPotInfo.srv | 设置花盆信息 |
| database | /pot/get | GetPotInfo.srv | 获取花盆信息 |
| database | /pot/delete | DeletePot.srv | 删除花盆 |

#### Published Topics

表8 database发布话题表

|  |  |  |  |
| --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 |
| database | /pot/update | PotUpdate.msg | 发送更新/删除的pot id |

### map\_provider

#### Published Topics

表9 map\_provider发布话题表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 订阅者 |
| map\_provider | /map | Map.msg | 提供地图信息 | navigation |

* 名称省略前缀 /map\_provider/msg/

#### Services

表10 database Services通信表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 调用者 |
| map\_provider | save\_map | SaveMap.srv | 保存地图数据 | manual\_map  auto\_map |
| map\_provider | clear\_map | ClearMap.srv | 删除地图数据 | manual\_map  auto\_map |

* 名称省略前缀 /map\_provider/
* 类型省略前缀 /map\_provider/srv/

#### Actions

表11 database Actions通信表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 调用者 |
| map\_provider | auto\_init\_map/goal | InitMap.action | 开启自动建图 | auto\_map |
| auto\_init\_map/cancel | 取消自动建图 |
| auto\_init\_map/feedback | 反馈自动建图进度 |
| auto\_init\_map/status | 建图 goal 状态 |
| auto\_init\_map/result | 返回自动建图结果 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 调用者 |
| map\_provider | manual\_init\_map/goal | InitMap.action | 开启手动建图 | manual\_amp |
| manual\_init\_map/cancel | 取消手动建图 |
| manual\_init\_map/feedback | 反馈手动建图进度 |
| manual\_init\_map/status | 建图 goal 状态 |
| manual\_init\_map/result | 返回手动建图结果 |

* 名称省略前缀 /map\_provider/
* 类型省略前缀 /map\_provider/action/

### object detect

#### Published Topics

表12 object\_detect 发布话题表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 订阅者 |
| object\_detect | pot\_abstract\_list | /object\_detect/pot\_abstrct\_list.msg | 输出所有花盆的摘要 | core |

#### Services

表13 object\_detect Services通信表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 调用者 |
| object\_detect | get\_pot\_info | /object\_detect/srv/GetPotInfo.srv | 获取指定花的信息 | core |
| object\_detect | set\_active | /object\_detect/srv/SetActive.srv | 设置指定花的active信息 | core |

### navigation

#### Actions

表14 navigation Actions通信表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 调用者 |
| navigation | goal | Navigate.action | 开始导航 | core |
| cancel | 取消特定ID的导航 |
| feedback | 反馈导航进度 |
| status | 导航 goal 状态 |
| result | 返回导航结果 |

* 名称省略前缀 /navigation/navigate/
* 类型省略前缀 /navigation/action/

### jet

#### Actions

表15 jet Actions通信表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点 | 名称 | 类型 | 作用 | 调用者 |
| arm | goal | Aim.action | 开始浇水 | target  auto\_water |
| cancel | 取消浇水 |
| feedback | 反馈浇水进度 |
| status | 浇水 goal 状态 |
| result | 返回浇水结果 |

* 名称省略前缀 /jet/
* 类型省略前缀 /jet/action/

# 详细设计

## 前端

### 视图层View

#### 顶栏

顶栏显示项目名称, 以及提供模块切换功能, 包含通知和关于按钮。



图11 顶栏设计图

#### 花盆信息侧栏

花盆信息侧栏展示花盆信息

* 切换展示方式
* 用户可以选择是否激活该花盆
* 点击定位以在地图上定位到该花盆位置



图12 侧栏设计图

#### 地图显示界面

地图展示界面显示3D 地图, 并包含地图相关选项。

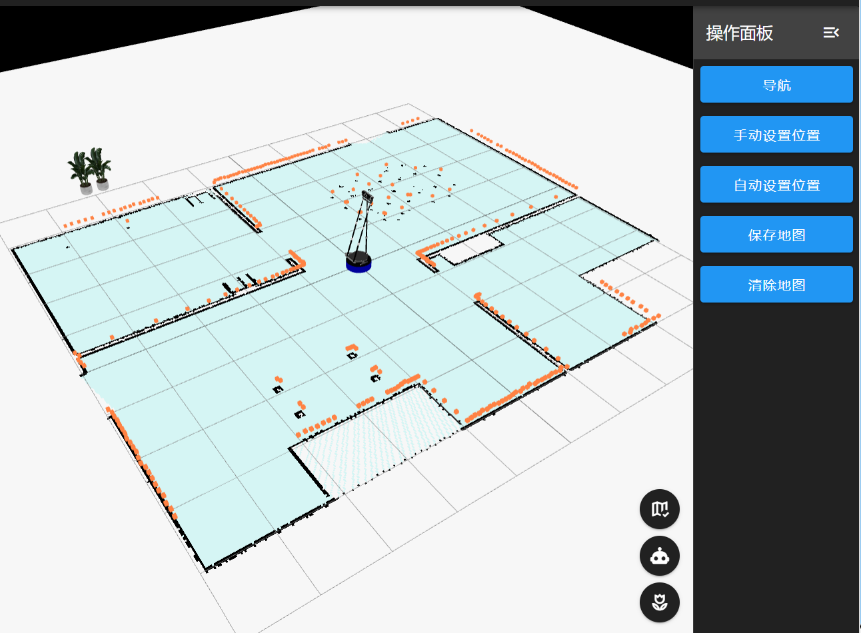


图13 地图显示栏设计图

#### 模式相关操作侧栏

针对不同模式提供不同操作。见系统用户界面设计。

#### 底栏

底栏显示 ROS 连接状态, 当前模式工作状态, 以及急停按钮。



图14 底栏设计图

### 数据层 Model

* global-map：

全局地图信息。维护栅格地图信息, 更新全局地图实例。

* robot：

机器人信息。维护机器人模型。

* lidar：

激光雷达信息。维护激光雷达点阵信息。

* navigation：

提供导航功能, 负责与 bridge 交互, 维护导航路径信息。

* camera-control：

负责维护相机位置, 处理自动跟随事件和定位花盆事件。

* object-detect：

检测花盆, 维护花盆信息。

## 控制器

### core节点

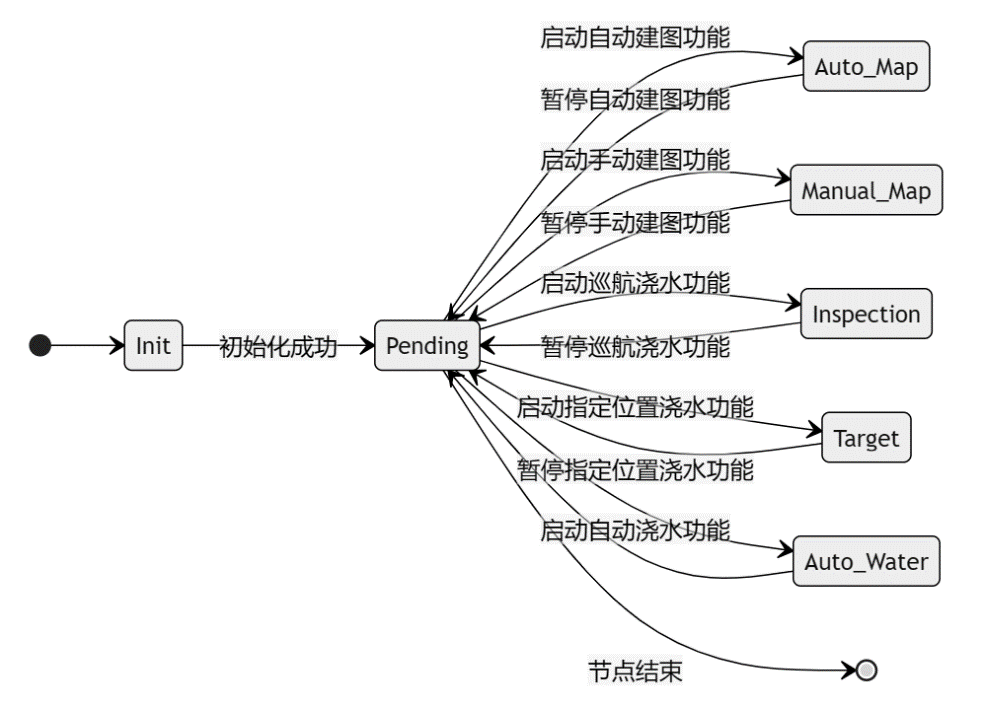


图15 core节点状态转移图

core 负责初始化, 管理全局状态, 以及通用的机器人操作。

* 切换全局状态
* 启停当前状态功能
* 切换急停

### pending 节点

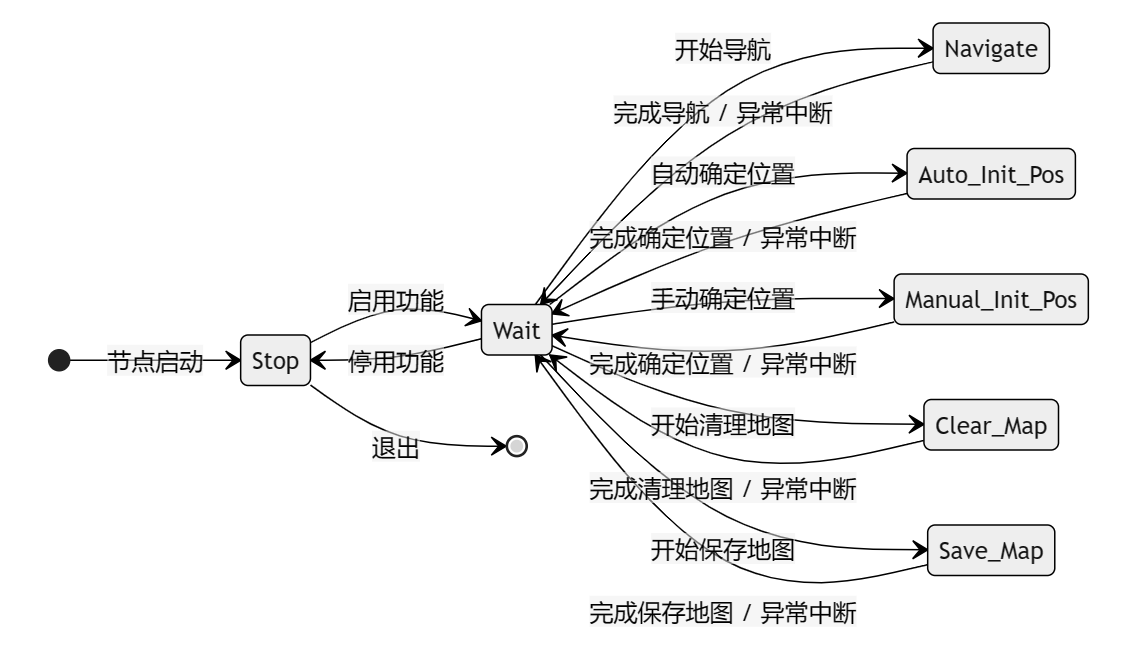


图16 pending节点状态转移图

pending在机器人无其他任务时启动, 可以对机器人的相关状态进行修改。

* 启停节点功能
* 清空地图
* 手动设置机器人初始位置
* 自动设置机器人初始位置
* 切换花盆激活状态

### auto\_map节点

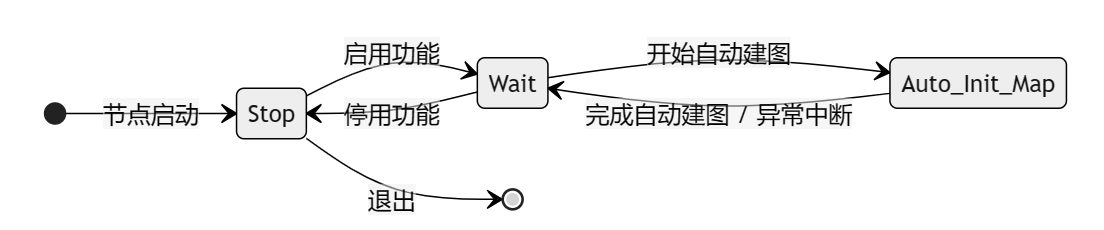


图17 auto\_map节点状态转移图

auto\_map负责调用建图模块进行自动建图, 提供控制功能。

* 启停节点功能

### manual\_map 节点

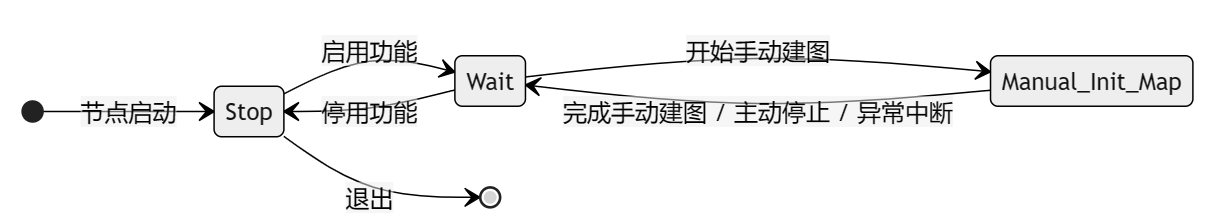


图18 manual\_map节点状态转移图

manual\_map 负责调用建图模块进行手动建图, 提供控制功能。

* 启停节点功能

### inspection 节点

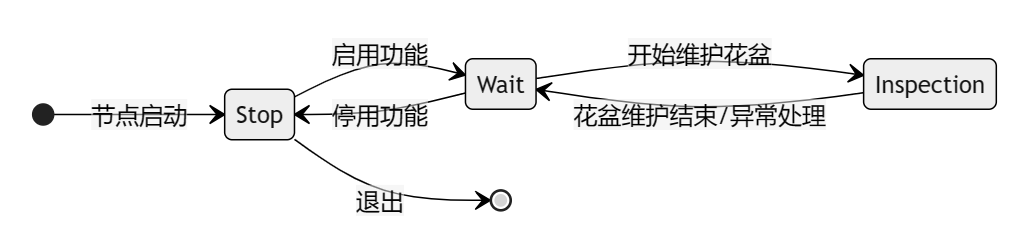


图19 inspection节点状态转移图

inspection负责检查数据库中的花盆是否仍处在原先的位置上，维护数据库信息和温室花盆位置数据的一致性。

* 启停节点功能

### target 节点

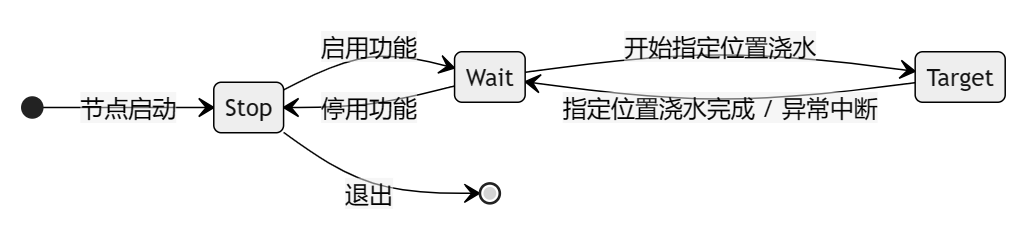


图20 target节点状态转移图

target 负责指定位置浇水, 提供控制功能。

* 启停节点功能
* 设置浇水花盆列表

### auto\_water 节点

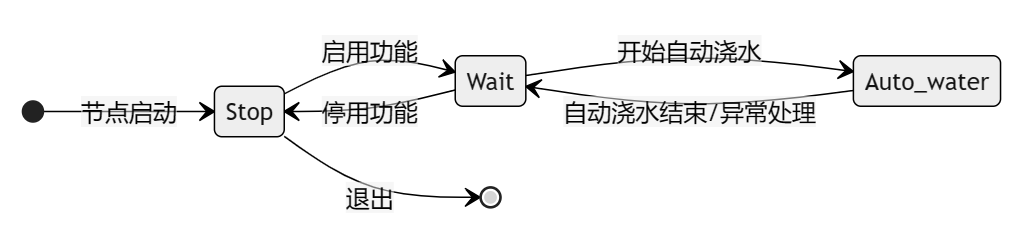


图21 auto\_water节点状态转移图

auto\_water 负责自动巡检浇水, 提供控制功能。

* 启停节点功能

## 建图

### 自动建图

自动建图是本浇花机器人项目的一项核心功能。map\_provider会提供/map\_provider/auto\_init\_map动作服务器，接受InitMap.action请求。在接收到以goal\_id作为唯一标识的建图任务后，系统会启动自动建图流程，通过Action接口调用/map\_provider/auto\_init\_map/goal，机器人随即启动Gmapping SLAM算法进行实时建图。

在建图过程中，机器人通过集成的激光雷达捕捉环境数据，实时生成点云信息。Gmapping SLAM算法整合这些数据，同时处理来自机器人内部传感器的信息，实时估算机器人的位置和姿态，并据此构建出场景的三维地图。

建图进度和建图结果通过/map\_provider/auto\_init\_map/feedback主题发布，并用户界面实时反馈建图信息，比如已完成建图的区域面积、地图覆盖率、建图质量评估等。

建图完成后，系统通过/map\_provider/auto\_init\_map/result主题发布完整的地图构建结果，整个地图数据会被妥善保存至本机，并同步更新到系统内部的地图数据结构，以便用于后续的导航、避障及任务规划。

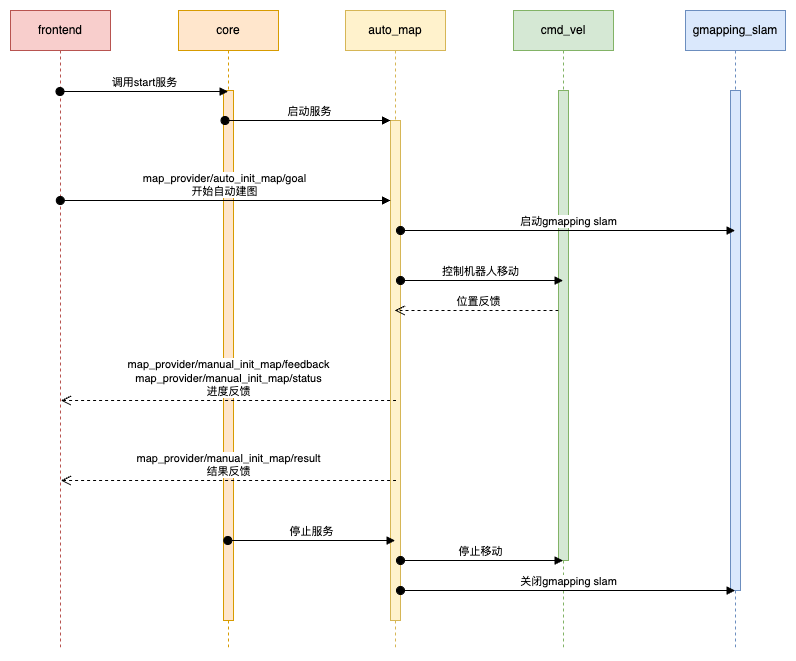


图22 自动建图时序图

### 手动建图

系统提供/map\_provider/manual\_init\_map Action，接受InitMap.action请求。该模式类似于自动建图，同时可能会根据用户指令控制机器人的移动和扫描行为，实现更精细的建图。该模式接收和处理/map\_provider/manual\_init\_map/goal和/map\_provider/manual\_init\_map/cancel请求，允许即时中断，同时确保在手动建图期间取消任何正在进行的自动建图任务。

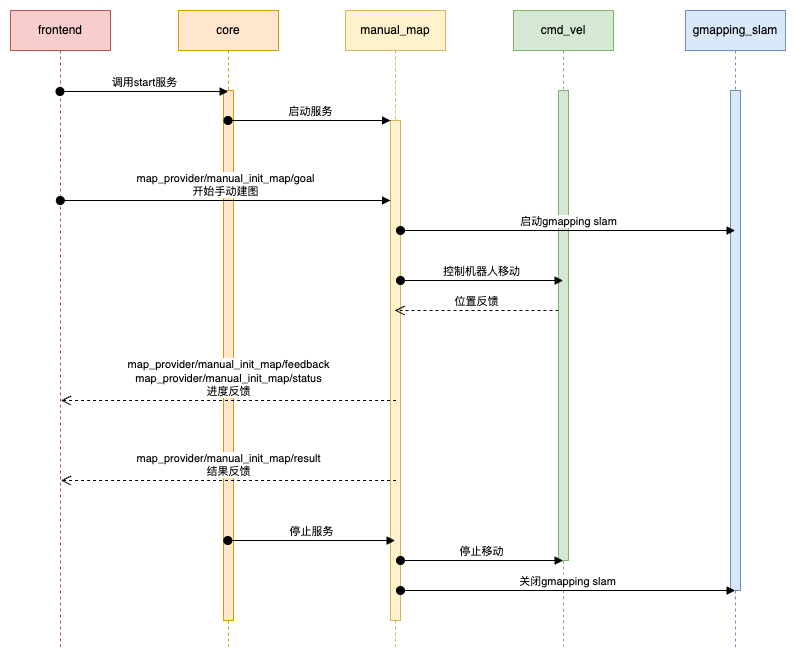


图23 手动建图时序图

### 保存地图

系统会订阅 /map\_provider/save\_map主题，当用户通过前端界面发起保存地图指令时，实际上最终会向该主题发布一条SaveMap.msg消息。map\_provider节点收到此消息后，将当前实时地图的状态转换为合适的数据格式（如.pgm和.yaml），并将这些文件保存到本机，以便于后续加载使用。

在启动阶段，或在需要加载已保存地图的时候，通过调用相应的加载服务，系统能够从文件系统中读取地图数据，并将其发布到/map主题，使整个系统可以快速加载并使用预先构建好的地图。

### 位置初始化

位置初始化是机器人在温室环境中开始执行任务前的重要步骤，它涉及到机器人在未知环境中的自我定位以及在已知环境中的精确复位。在这个环节中，本设计采用蒙特卡洛算法来辅助机器人自动探测并确认自身的初始位置。

当建图完成之后，系统会调用/pending/auto\_init\_pos服务，触发机器人使用传感器数据和蒙特卡洛方法随机采样多个可能的位置，计算与已构建地图中特征点的匹配度，从而确定机器人在地图中的位置。前端也可以主动触发该服务，提供直观的操作界面，允许用户点击按钮启动自动位置初始化过程。

### 位置校准

位置校准模块提供人工介入机器人系统的机会，以确保机器人在复杂或多变的温室环境下仍然能够准确无误地知道自己的位置。

在地图显示界面上，用户可以在三维地图上直接点击期望的初始位置，此时前端调用 /pending/manual\_init\_pos服务，系统会立即将新坐标信息同步至机器人控制系统。/pending/manual\_init\_pos服务在接受到用户的手动校准请求后，会进行必要的验证，如检查所选位置是否合理，避免机器人碰撞或进入不可通行区域等。校准成功后，系统会记录此次校准事件及其关联位置信息，并向前端反馈以更新地图的显示数据，以备后续导航和任务执行时使用。

### 清空地图

为了支持重新建图或清除不准确的环境数据，系统提供了一个地图清空功能。系统订阅/map\_provider/clear\_map主题，当用户发出清空地图的指令时，系统最终会接收到ClearMap.msg消息，而后，系统会清空当前内存中的地图，并发布一个新的空白地图至/map主题。同时，如有必要，系统也会清理相关的数据库记录，确保新的建图数据不会受到旧数据的影响。

## 导航与动态避障

### 导航系统分层

首先，core节点是该项目的控制节点。它可以启动三个子控制节点来分别实现定点浇水（target），花盆维护（inspection），自动巡航浇水（auto\_water）三种服务。

simple\_goal封装了move\_base节点，实现了定点导航和动态避障功能。

我们将导航节点分解出以下四个层次。

core节点在控制层，决定启动哪一种浇水模式，并将具体的服务任务交付给服务层。

target，inspection，auto\_water是服务层的节点，封装了各自的导航逻辑，分别调用定点导航节点simple\_goal。

simple\_goal是功能层的定点导航节点，是move\_base的上层封装。我们的动态避障功能也是在此实现。

move\_base是导航的官方包，是导航行为的具体实现层。

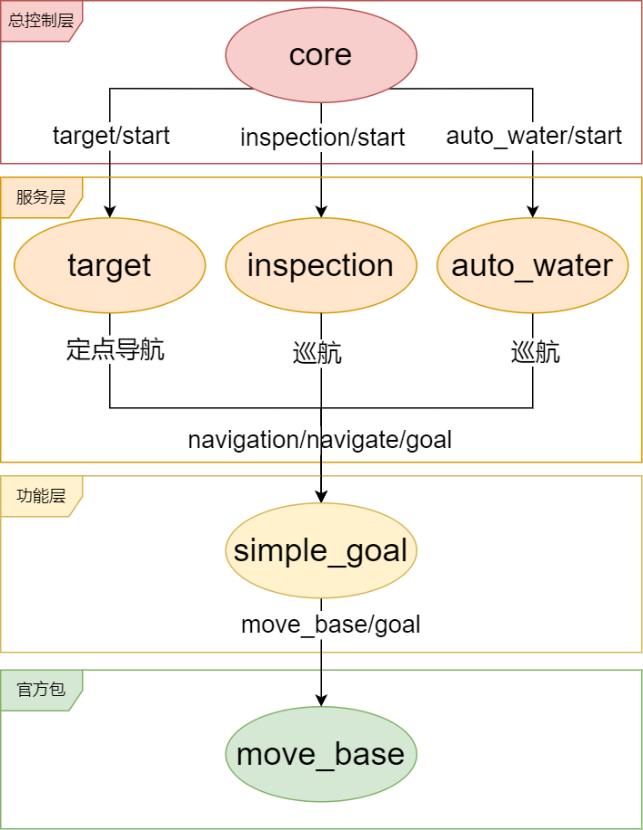


图24 导航系统分层图

### 定点浇花

1. frontend调用core节点的start服务，start服务调用target/start服务，启动定点浇水服务节点target。

2. frontend选定目标花盆，将位置发送到话题target/set\_target/goal中。

3. target节点订阅该话题，将航点通过话题navigation/navigate/goal转发给simple\_goal节点。

4. simple\_goal把航点通过move\_base/goal话题转发给move\_base节点。

5. move\_base节点根据代价地图等信息进行导航。

6. 到达航点后，target节点收到该消息，启动浇花功能。

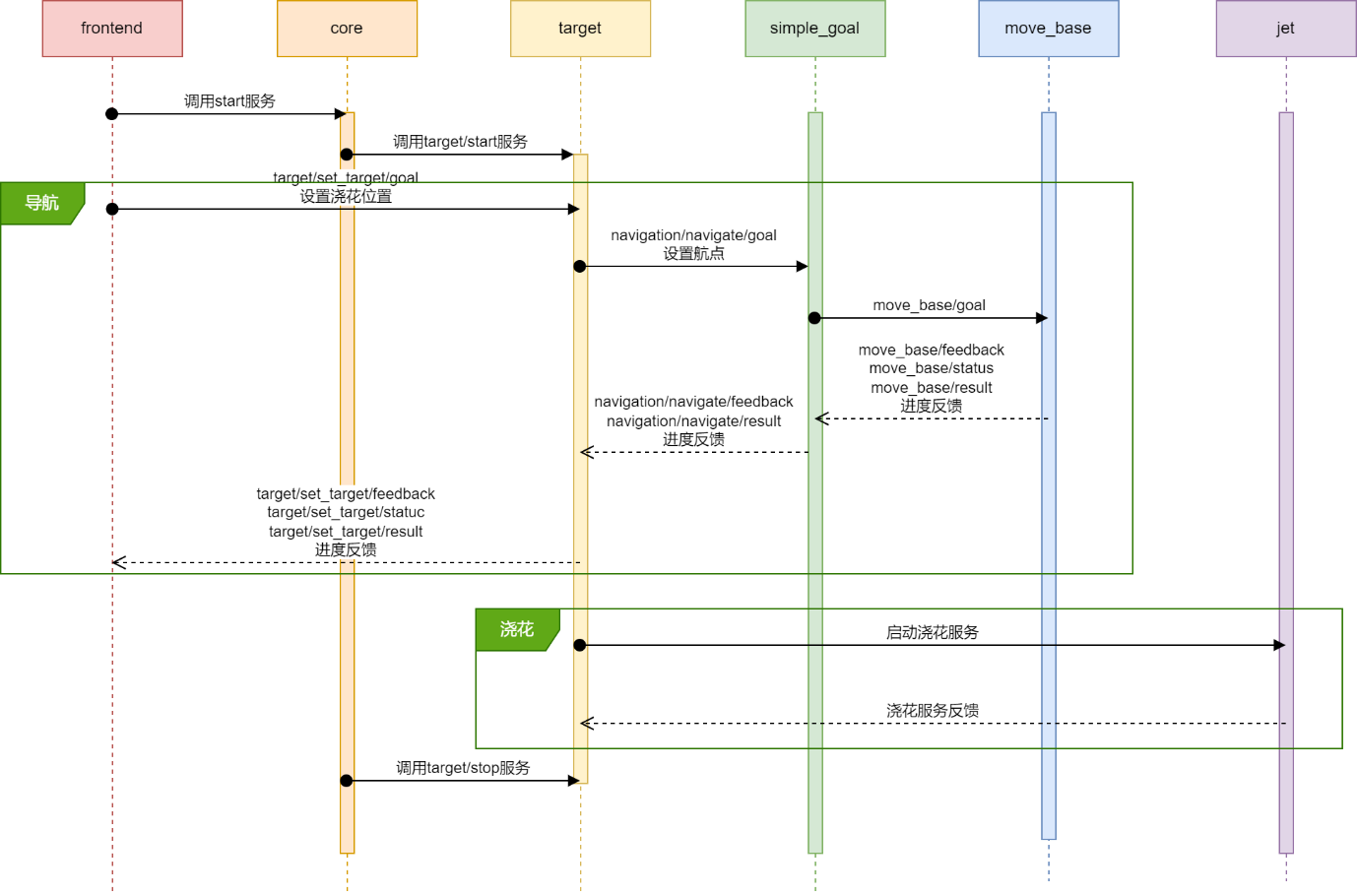


图25 定点浇花时序图

### 自动巡航浇花

该例程实现逻辑类似于定点浇花，通过多次向simple\_goal节点发送航点即可实现。缺水巡检模式的实现逻辑与自动巡航浇花的逻辑类似。

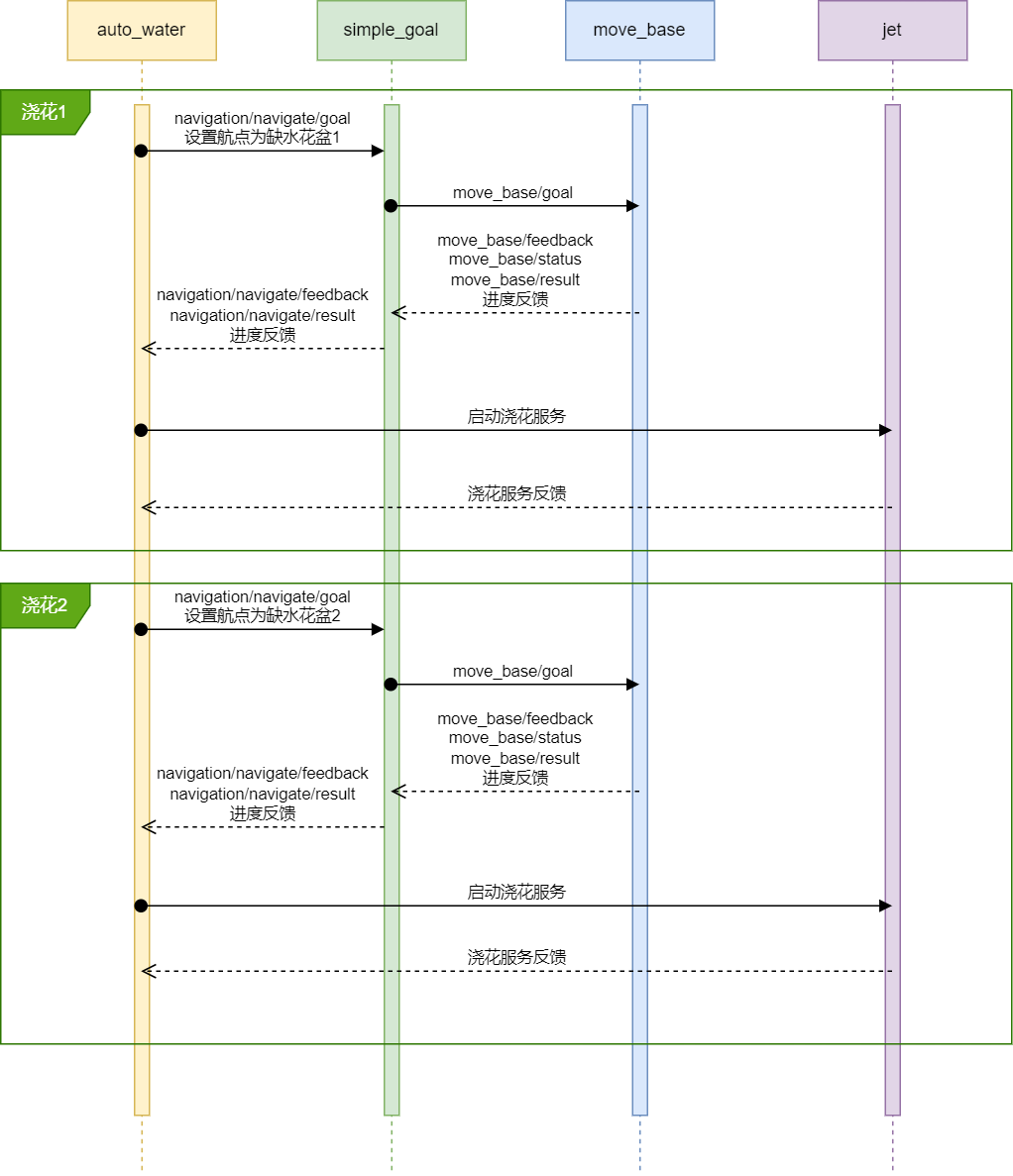


图26 自动巡航浇花时序图

### 动态避障实现

动态避障实现的基本思想是simple\_goal节点向move\_base发送航点，同时订阅/\_cmd\_vel话题。若持续一段时间内速度过低，则可断言遇到障碍，此时simple\_goal节点重新发送航点规划路径，从而实现动态避障。

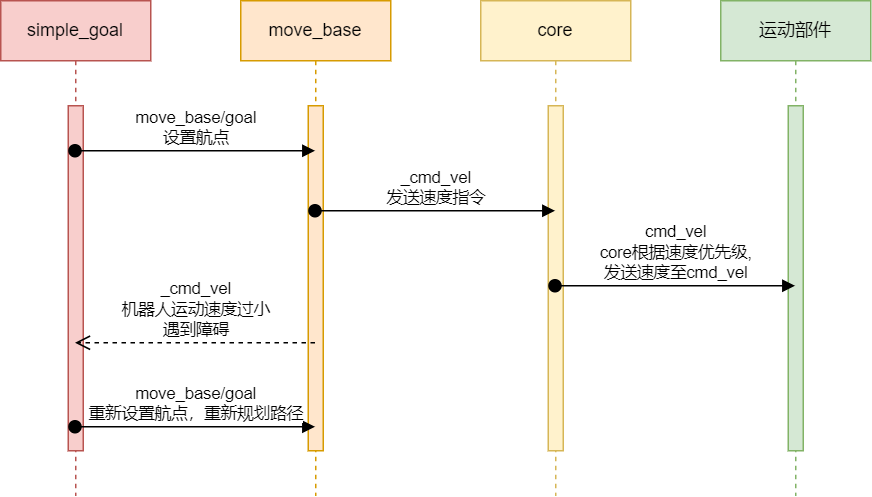


图27 动态避障时序图

## 花盆识别

花盆识别部分的时序图如下图所示，在本节中的后续部分将对该时序图中花盆识别部分的核心设计细节进行分析与阐述。本项目的一大亮点在于将YOLO模型部署到ROS嵌入式系统上，以实现高精度、低延时的实时目标检测。

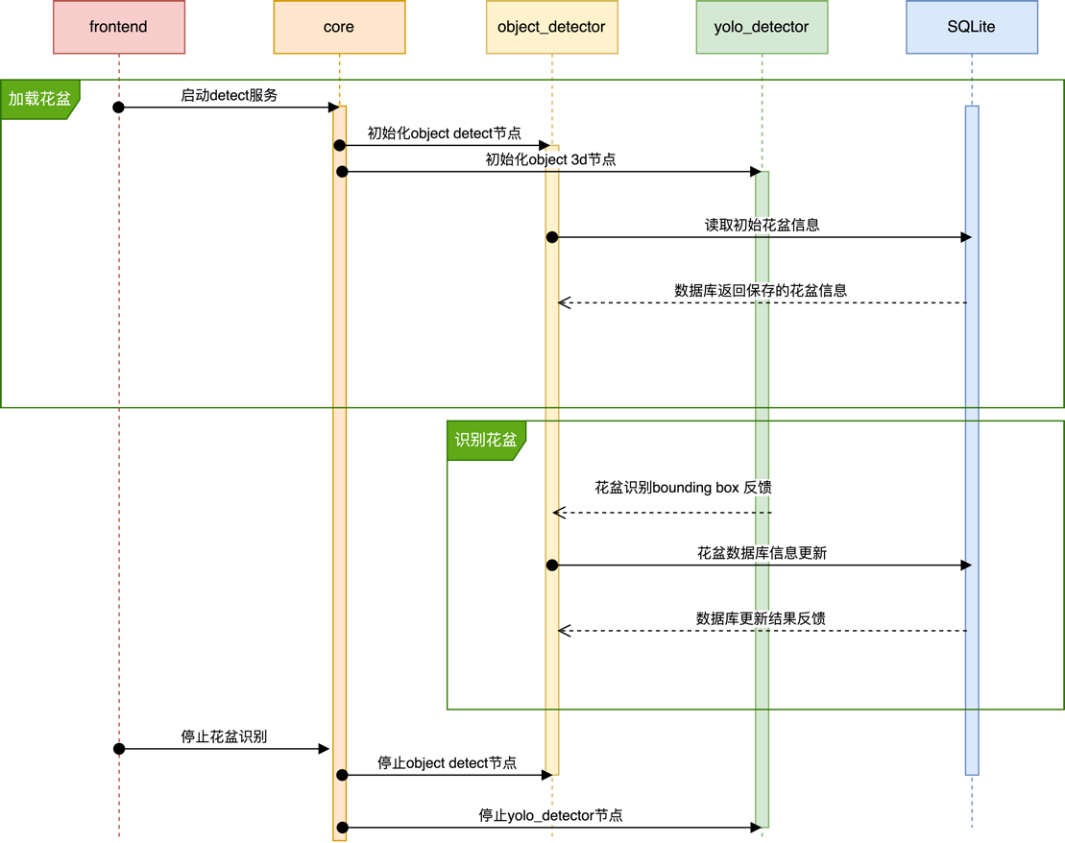


图28 花盆识别时序图

### 2D图片识别节点yolo\_detector

2D图片数据处理模块订阅的话题：

* /kinect2/hd/image\_color\_rect：来自Kinect摄像头的高清图像信息

2D图片数据处理模块发布的话题为：

* /yolo\_detector/BoundingBoxes：yolo detector在完成花盆识别后，将检测到的二维像素上的检测框坐标发出。

节点总体的识别流程如下：（1）初始化节点和订阅话题：YoloDetector节点初始化后，订阅来自Kinect摄像头的高清图像话题（/kinect2/hd/image\_color\_rect），以及速度（/cmd\_vel）和位置（/amcl\_pose）话题。（2）加载模型和参数：加载YOLO模型的权重文件和相关参数，包括模型置信度阈值、是否可视化以及发布速率等。

图像数据处理：接收到图像数据后，将ROS图像消息转换为NumPy数组，并根据需要进行颜色空间转换，以适应YOLO模型的输入格式。（3）目标检测：使用YOLO模型对图像进行目标检测，获取检测框的坐标和置信度信息。（4）结果发布：将检测到的目标信息封装成BoundingBoxes消息，并通过预定义的话题（/yolo\_detector/BoundingBoxes）发布。同时，若开启可视化选项，检测结果将显示在图像窗口中。（5）图像发布：将包含检测框的处理后图像转换回ROS图像消息，并通过指定话题（/yolo\_detector/detection\_image）发布。

### 花盆识别节点 object\_detector节点

花盆识别功能包中的核心为花盆识别节点object\_detector，负责进行花盆信息管理、控制信息接收等重要功能。

节点初始化时，从数据库加载花盆信息，并订阅多个话题，包括速度信息、相机图像、YOLO检测结果、深度图像和相机内参。使用message\_filters对这些话题进行时间同步处理。在接收到同步的消息后，节点通过YOLO检测结果和深度图像计算花盆的三维坐标，并将其转换到地图坐标系下。检测到的花盆位置被存储和更新，若是新花盆则添加到数据库中，并发布检测到的花盆中心点信息。节点还提供服务接口，用于检查和更新花盆信息。

## 浇花

浇花模块扮演着工具包的角色，直接由auto\_water和target节点调用，其工作时序图如下图所示。

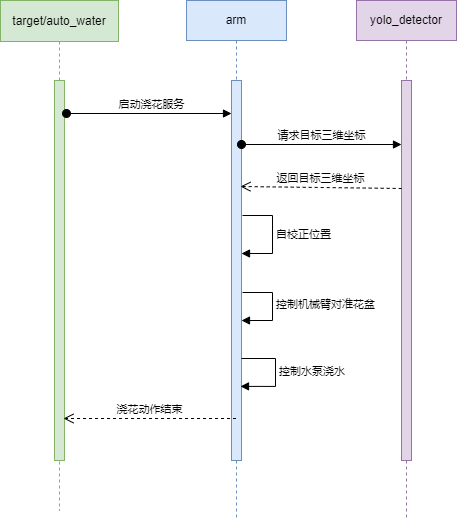


图29 浇花动作时序图

### 机械臂控制

机械臂负责移动喷水软管到指定的浇花位置。当机械臂控制喷水软管对准花盆后，会先后执行“位置矫正”和“开闭水泵”动作。

### 位置矫正

机械臂接收发自YOLO detector的花盆三维坐标数据，并根据Y方向（机器人侧向）偏移进行旋转矫正；依据X方向（机器人正对方向）偏移进行前后平移矫正。

### 水泵控制

水泵控制功能主要由单片机和继电器共同实现。单片机与上位机通过串口连接，接受来自arm节点的消息，并控制继电器的开闭状态。当继电器处于开启状态时，水泵通电抽水。经过短暂浇水时间间隔，继电器关闭，水泵断电结束浇花动作。

## 缺水识别

### 曲线拟合与数据分析

由于本项目的主要应用场景是温室等较为密闭的空间中，局部环境较为稳定。因此，本项目的缺水识别功能采用指数衰减模型来拟合土壤湿度随时间的变化，将植物分为幼苗期、生长期和花期，每个阶段使用不同的曲线模型[1]（类似于sigmoid函数）估计植物水分。在实际的应用过程中，可以需要根据不同盆栽植物的实际情况(例如温室中种植的植物、当前植物的生长阶段)设置适合当前环境的指数衰减模型参数，并可以根据R2等指标来评估拟合的效果。

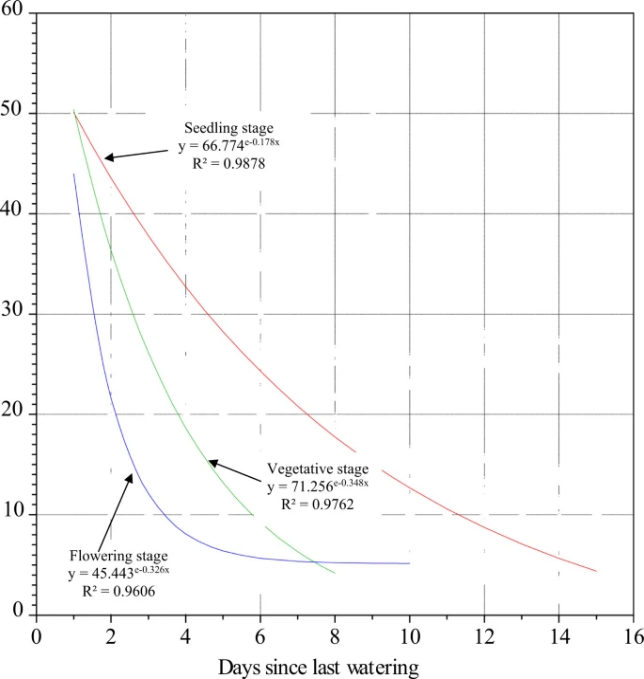


图30 土壤湿度随时间变化曲线图[1]

### 缺水识别逻辑

缺水识别模块的核心识别逻辑是根据“距离植物上次浇水的时间差”和“预设的指数衰退模型”计算当前植物的土壤湿度含量，并根据缺水阈值判断植物是否进入缺水状态。若植物被识别为缺水，“自动巡检浇水”例程会自动前往该植物处进行浇水，及时为缺水植物补充水分。

# 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

# 需求可追踪性说明

表16 需求可追踪说明表

|  |  |
| --- | --- |
| SRS文档功能/非功能需求 | SDD文档体系结构设计 |
| 建图 | 6.3 建图 |
| 地图信息展示 | 6.1 前端 |
| 导航 | 6.4 导航与动态避障 |
| 花盆识别 | 6.5 花盆识别 |
| 缺水识别 | 6.7 缺水识别 |
| 浇花 | 6.5 浇花 |
| 异常处理 | 6 详细设计 |
| 响应时间 | 4.3 支撑体系结构 |
| 功耗 | 4.3 支撑体系结构 |
| 处理能力 | 6 详细设计 |
| 系统可用性 | 6 详细设计 |
| 可移植性 | 4.1 软硬件体系结构 |
| 完整性 | 4体系结构设计5接口设计 6详细设计 |
| 效率 | 4体系结构设计 6详细设计 |
| 健壮性 | 4体系结构设计 6详细设计 |
| 安全性 | 4体系结构设计 6详细设计 |
| 维护性 | 4.3 支撑体系结构 |
| 合规性 | 4体系结构设计 6详细设计 |