**浇花机器人——温室管家**

**软件测试计划**

**STP-03**

**V2.0**

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 |
| 2 | Hao7un | 主编、小组讨论、计划、测试进度表、评价 |
| 5 | Mxode | 审核、小组讨论、计划 |
| 9 | LTT | 小组讨论、计划、测试进度表 |
| 1 | Le | 小组讨论、范围、任务概述、软件测试环境、计划、评价 |
| 6 | Kun | 小组讨论、计划、各需求对应的测试 |

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 | 2024.5.13 | Hao7un | Mxode | STP初始版本 |
| V1.1 | 2024.5.16 | Hao7un | Mxode | 规范测试用例 |
| V1.2 | 2024.5.18 | Hao7un | Mxode | 细化白盒测试的分支覆盖 |
| V1.3 | 2024.5.26 | Hao7un | Mxode | 增加白盒测试状态图 |
| V1.4 | 2024.5.27 | Hao7un | Mxode | 完善系统性能测试用例 |
| V1.5 | 2024.5.28 | Hao7un | Mxode | 白盒测试状态图路径分析 |
| V2.0 | 2024.6.7 | Hao7un | Mxode | 测试文档一致性 |

目录

[1. 范围 1](#_Toc168649701)

[1.1 项目概述 1](#_Toc168649702)

[1.1.1 背景 1](#_Toc168649703)

[1.1.2 主要功能 1](#_Toc168649704)

[1.1.3 非功能需求 2](#_Toc168649705)

[1.1.4 应用场景 3](#_Toc168649706)

[1.2 文档概述 3](#_Toc168649707)

[1.2.1 用途 3](#_Toc168649708)

[1.2.2 内容组织 4](#_Toc168649709)

[1.3 术语和缩略词 4](#_Toc168649710)

[1.3.1 术语 4](#_Toc168649711)

[1.3.2 缩略词 5](#_Toc168649712)

[1.4 引用文档 5](#_Toc168649713)

[2. 任务概述 6](#_Toc168649714)

[2.1 数据需求 6](#_Toc168649715)

[2.2 功能需求 7](#_Toc168649716)

[2.3 异常处理需求 11](#_Toc168649717)

[2.4 业务需求 13](#_Toc168649718)

[2.5 非功能需求 17](#_Toc168649719)

[3. 软件测试环境 18](#_Toc168649720)

[3.1 Gazebo仿真环境 18](#_Toc168649721)

[3.1.1 软件项 18](#_Toc168649722)

[3.1.2 硬件及固件项 19](#_Toc168649723)

[3.1.3 其它材料 19](#_Toc168649724)

[3.1.4 参与组织 20](#_Toc168649725)

[3.1.5 测试人员 20](#_Toc168649726)

[3.2 实验室真实环境 20](#_Toc168649727)

[3.2.1 软件项 20](#_Toc168649728)

[3.2.2 硬件及固件项 21](#_Toc168649729)

[3.2.3 其它材料 21](#_Toc168649730)

[3.2.4 参与组织 22](#_Toc168649731)

[3.2.5 测试人员 22](#_Toc168649732)

[4. 计划 23](#_Toc168649733)

[4.1 总体设计 23](#_Toc168649734)

[4.1.1 测试级别划分 23](#_Toc168649735)

[4.1.2 测试类别划分 25](#_Toc168649736)

[4.1.3 一般测试条件 27](#_Toc168649737)

[4.1.4 测试过程 28](#_Toc168649738)

[4.1.5 测试结果记录规范 30](#_Toc168649739)

[4.2 计划执行的测试 37](#_Toc168649740)

[4.2.1 单元测试 37](#_Toc168649741)

[4.2.1.1 单元测试目标 37](#_Toc168649742)

[4.2.1.2 单元测试用例 37](#_Toc168649743)

[4.2.1.3 单元测试计划 39](#_Toc168649744)

[4.2.2 集成测试 40](#_Toc168649745)

[4.2.2.1 集成测试目标 40](#_Toc168649746)

[4.2.2.2 集成测试用例 40](#_Toc168649747)

[4.2.2.3 集成测试计划 41](#_Toc168649748)

[4.2.3 系统测试 41](#_Toc168649749)

[4.2.3.1 系统测试目标 41](#_Toc168649750)

[4.2.3.2 系统测试用例 41](#_Toc168649751)

[4.2.3.3 系统测试计划 42](#_Toc168649752)

[4.3 测试用例 43](#_Toc168649753)

[4.3.1 单元测试：机器人 43](#_Toc168649754)

[4.3.1.1 移动模块单元测试 43](#_Toc168649755)

[4.3.1.2 手柄单元测试 46](#_Toc168649756)

[4.3.1.3 雷达单元测试 49](#_Toc168649757)

[4.3.1.4 摄像头单元测试 51](#_Toc168649758)

[4.3.1.5 扬声器单元测试 54](#_Toc168649759)

[4.3.1.6 机械臂单元测试 55](#_Toc168649760)

[4.3.2 单元测试：前端交互界面 57](#_Toc168649761)

[4.3.2.1 顶栏功能单元测试 57](#_Toc168649762)

[4.3.2.2 花盆列表单元测试 59](#_Toc168649763)

[4.3.2.3 3D地图单元测试 61](#_Toc168649764)

[4.3.2.4 控制面板单元测试 66](#_Toc168649765)

[4.3.2.5 底栏单元测试 74](#_Toc168649766)

[4.3.3 单元测试：ROS端测试 77](#_Toc168649767)

[4.3.3.1 controller Package 77](#_Toc168649768)

[4.3.3.2 navigation Package 101](#_Toc168649769)

[4.3.3.3 map\_provider Package 106](#_Toc168649770)

[4.3.3.4 object\_detect Package 110](#_Toc168649771)

[4.3.3.5 robot\_arm Package 119](#_Toc168649772)

[4.3.3.6 pot\_database Package 125](#_Toc168649773)

[4.3.4 集成测试：功能需求 129](#_Toc168649774)

[4.3.4.1 建图 129](#_Toc168649775)

[4.3.4.2 导航与动态避障 132](#_Toc168649776)

[4.3.4.3 浇水 135](#_Toc168649777)

[4.3.4.4 启动服务 138](#_Toc168649778)

[4.3.5 系统测试：业务需求 141](#_Toc168649779)

[4.3.5.1 自动建图测试 141](#_Toc168649780)

[4.3.5.2 手动建图测试 146](#_Toc168649781)

[4.3.5.3 巡检维护测试 151](#_Toc168649782)

[4.3.5.4 定点浇水测试 153](#_Toc168649783)

[4.3.5.5 自动巡检浇水测试 164](#_Toc168649784)

[4.3.5.6 等待服务测试 175](#_Toc168649785)

[4.3.5.7 系统性能测试 183](#_Toc168649786)

[5. 测试进度表 187](#_Toc168649787)

[5.1 测试时间安排 187](#_Toc168649788)

[5.2 进度约束 189](#_Toc168649789)

[5.2.1 测试阶段进度约束 189](#_Toc168649790)

[5.2.2 测试样例进度约束 189](#_Toc168649791)

[5.2.3 关键路径法分析 190](#_Toc168649792)

[6. 各需求对应的测试 192](#_Toc168649793)

[6.1 数据需求 192](#_Toc168649794)

[6.2 功能需求 192](#_Toc168649795)

[6.3 异常处理需求 193](#_Toc168649796)

[6.4 业务需求 193](#_Toc168649797)

[6.5 非功能需求 193](#_Toc168649798)

[7. 评价 194](#_Toc168649799)

[7.1 评价准则 194](#_Toc168649800)

[7.2 数据处理 195](#_Toc168649801)

[7.3 结论 195](#_Toc168649802)

# 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、动态避障等技术，实现缺水巡检、自动浇灌等功能，从而降低人工成本并提高温室管理的智能化水平，适用于温室农作物的精细化照料。

### 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。

### 非功能需求

* 可靠性与稳定性：机器人必须保证连续不间断的运行能力，即使在连续长时间工作或复杂的温室环境下也能保持高可靠性与稳定性。
* 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。

### 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

本文档的用途如下：

* 根据需求制定测试内容：本文档基于SRS-软件需求规格说明书，围绕“**数据需求**”、“**功能需求**”、“**异常处理需求**”、“**业务需求**”和“**非功能需求**”开展测试。每个测试环节明确需要测试的对象，保证测试全面覆盖。
* 明确测试环境与人员：明确测试环境配置，包括机器人实验室和Gazebo仿真环境的软硬件设备和材料。同时明确测试人员任务分配，确保测试团队分工明细。
* 定义测试级别与测试类型：将测试级别划分为“**单元测试**”“**集成测试**”“**系统测试**”以及后续的“**回归测试**”，将测试类别划分为“功能测试”和“性能测试”。
* 测试规范记录：制定规范测试的结果记录标准，如用例执行情况、黑盒测试结果，确保测试质量与可追踪。
* 测试样例设计：设计各类测试用例，包括“单元测试样例”、“集成测试样例”以及“系统测试样例”，确保从模块单元到系统级功能的全方位验证。
* 进度安排与跟踪：制定测试时间表，包括时间安排与约束，确保测试按期执行，同时对应需求跟踪，确保需求可以被测试完全覆盖。
* 反馈与报告：汇总测试过程中出现的问题，分析测试结论，为项目持续改进提供依据。

### 内容组织

本STP的内容组织如下：

* 项目概述
* 任务概述
* 软件测试环境
* 计划
* 测试进度表
* 需求的可追踪性
* 评价

## 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++: 高级编程语言

Python：高级编程语言

Rviz：机器人可视化平台

Gazebo：机器人仿真平台

Visual Studio Code：代码开发工具

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] SRS-03-软件需求规格说明书，V-2.1 2024-6.21

# 任务概述

根据SRS-需求规格说明书[1]，本项目的需求可划分为五大核心领域：（1）“**数据需求**”确保数据信息的准确无误与完整性；（2）“**功能需求**”关注各项核心服务执行的精确性；（3）“**异常处理需求**”关注对异常情况的处理应对机制；（4）“**业务需求**”确保每项需求分析中的系统级需求的正常运行；（5）“**非功能需求**”要求开发人员应充分考虑机器人系统的运作性能与稳定性。测试的所有环节都基于项目需求展开。

## 数据需求

在浇花机器人系统中，数据直接关联到系统对植物生长的精细化管理。具体而言，数据需求重点关注“植物信息”和“历史任务”两大数据类别。“植物信息”涵盖了每一个温室花盆的详细档案，对维护每一株植物的生长有至关重要的作用；“历史任务”记录了系统执行过的所有巡检与浇花任务记录，在本项目中充当日志的角色。围绕“数据需求”，本文计划对数据库接口进行充分测试，保障浇花机器人系统的数据质量。

**表2 数据需求表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 需求编号 | 数据需求实体 | 简要描述 | 需求实现单元 | 是否测试 |
| 1 | 植物信息 | 包含序号、三维坐标、最近浇水时间、点云等信息。 | 数据库 | 是 |
| 2 | 历史任务 | 包含序号、任务时间、任务模式、完成情况信息。 | 数据库 | 是 |

## 功能需求

本项目的功能需求主要围绕ROS机器人功能和用户前端界面展开。ROS机器人系统主要包括“手动建图”、“自动建图”、“导航与动态避障”、“花盆识别”、“浇花”等功能。用户可通过用户前端界面向机器人系统发送指令，调用机器人系统提供的相关功能模块；机器人可将功能执行结果实时返回并显示在用户前端界面上，或者以语音提示的方式提醒用户。

具体功能需求记录在“功能需求”[1]部分，其中需求编号1-6描述了有关ROS机器人的需求，需求编号7-13描述了有关用户前端界面的需求。各部分需求对应的实现单元都需要进行充分的测试。

**表3 功能需求表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 需求编号 | 功能需求 | 简要描述 | 需求实现单元 | 是否测试 |
| 1 | 手动建图 | 1. 用户在用户等待界面点击“手动建图”，激活手动建图模式； 2. 机器人开启雷达监测，捕捉周围环境信息； 3. 用户操纵手柄，引导机器人在温室内移动，使用激光雷达扫描周围环境； 4. 收集完温室内完整地图数据后，用户终止建图，手动建图结束。 | 用户界面  数据库单元  雷达单元  运动单元  手柄单元  总控模块 | 是 |
| 2 | 自动建图 | 1. 用户在用户等待界面点击“自动建图”，激活自动建图模式； 2. 机器人开启雷达监测，捕捉周围环境信息； 3. 机器人调用RRT算法在温室内建立路径搜索树，游走过程中使用激光雷达扫描周围环境； 4. 收集完所有路径信息后，机器人自行停止建图，自动建图结束。 | 用户界面  数据库单元  雷达单元  运动单元  总控模块 | 是 |
| 3 | 导航与动态避障 | 引导机器人走到目标花盆处。 | 用户界面  数据库单元  雷达单元  运动单元  总控模块 | 是 |
| 4 | 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。 | 数据库单元  相机单元  总控模块 | 是 |
| 5 | 浇花 | 机械臂抓取软管对准花盆，控制水泵通电浇水，并更新数据库中的花盆信息。 | 数据库单元  雷达单元  机械臂单元  水泵单元  语音输出单元  总控模块 | 是 |
| 6 | 数据库服务 | 管理植物在温室种的位置，含水量等数据和历史任务数据。 | 数据库单元 | 是 |
| 7 | 系统急停 | 应急情况下可让用户立即中断机器人运动。 | 用户界面  运动单元 | 是 |
| 8 | 维护功能 | 用户可检查系统当前版本，并将系统升级到最新版本。 | 用户界面  数据库  源码仓库 | 是 |
| 9 | 等待界面 | 提供多个可供切换的工作模式。 | 用户界面  数据库  总控模块 | 是 |
| 10 | 自动建图界面 | 显示温室3D地图，提供“自动跟随”等待选功能，“自动建图”模式选中高亮。 | 用户界面  数据库单元、  总控模块 | 是 |
| 11 | 手动建图界面 | 显示温室3D地图，提供“自动跟随”等待选功能，“手动建图”模式选中高亮。 | 用户界面  数据库单元  总控模块 | 是 |
| 12 | 指定位置浇水界面 | 显示温室3D地图，提供“自动跟随”等待选功能，“指定位置浇水”模式选中高亮。 | 用户界面  数据库单元  总控模块 | 是 |
| 13 | 巡检自动浇水界面 | 显示温室3D地图，提供“自动跟随”等待选功能，“巡检自动浇水”模式选中高亮。 | 用户界面  数据库单元  总控模块 | 是 |
| 14 | 系统信息展示 | 显示ROS连接状态、当前工作状态。 | 用户界面  总控模块 | 是 |

## 异常处理需求

异常处理对于确保系统稳定性和用户体验至关重要。本项目的异常处理需求如下表所示，这些需求确定了异常的识别以及处理流程、同时明确了相应的实现单元，确保每一项异常都能得到恰当的处理与反馈。

**表4 异常处理需求表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 需求编号 | 异常需求 | 简要描述 | 需求实现单元 | 是否测试 |
| 1 | 航点不可达 | 当用户选择地图上不可到达的航点时，主控将向用户界面返回“航点不可达”消息，提醒用户重新选择目标航点。 | 用户界面  总控模块  导航模块 | 是 |
| 2 | 花盆缺失异常 | 如果某个花盆因某些原因离开了原本的位置，机器人导航到该花盆处时将无法识别到目标花盆。此时浇水模块会向调用方反馈“花盆消失”信息，该异常最终被传递到用户界面。 | 用户界面  相机单元  总控模块  花盆识别模块  浇水模块 | 是 |
| 3 | 水泵缺水异常 | 如果水泵因为疏于补给水或漏水，致使无法为浇水模块提供充足的水源，浇水模块会向调用方返回“水泵缺水”异常，该异常最终被传递到用户界面。 | 用户界面  总控模块  浇水模块 | 是 |
| 4 | 建图不完全异常 | 如果建立的地图没有完全覆盖温室内的所有区域，可能会导致机器人在导航过程中遇到实际不可达的区域，阻碍正常的导航进程。此时导航模块会反馈“建图不完全”异常，该异常最终被传递到用户界面。 | 用户界面  总控模块  建图模块  导航模块  运动单元 | 是 |
| 5 | 系统启动异常 | 系统因缺失必要依赖导致无法启动。 | ROS软件环境  机器人硬件 | 是 |

## 业务需求

本项目的业务需求主要包括“手动建图”、“自动建图”、“定点浇水”和“自动巡检浇水”四类主要业务，和“巡检维护”，“等待服务”两类辅助业务，以满足高效且智能的精细化温室管理。业务需求描述如下表所示：

**表5 业务需求表**

|  |  |  |  |
| --- | --- | --- | --- |
| 需求编号 | 业务需求 | 简要描述 | 是否测试 |
| 1 | 自动建图 | 1. 用户启动机器人； 2. 用户界面切换至等待界面，提供“自动建图”候选模式； 3. 用户选择“自动建图”，机器人转入“自动建图”模式； 4. 机器人调用路径搜索算法自动在未知的温室区域游走，利用激光雷达扫描周围障碍物环境信息； 5. 自动建图完毕后，机器人停止建图，并保存温室环境地图； 6. 用户界面切换回最初的等待界面，机器人转回等待模式。 | 是 |
| 2 | 手动建图 | 1. 用户启动机器人； 2. 用户界面切换至等待界面，提供“手动建图”候选模式； 3. 用户选择“手动建图”，机器人转入“手动建图”模式； 4. 用户使用手柄操纵机器人在温室内部移动，利用激光雷达扫描周围障碍物环境信息； 5. 手动建图完毕后，用户手动停止建图，并保存温室环境地图； 6. 用户界面切换回最初的等待界面，机器人转回等待模式。 | 是 |
| 3 | 定点浇水 | 1. 用户启动机器人； 2. 用户界面切换至等待界面，提供“指定位置浇水”候选模式； 3. 用户选择“指定位置浇水”，机器人转入“指定位置浇水”模式； 4. 用户选择地图中的某个花盆作为目标航点； 5. 机器人调用导航模块功能，自主规划路径到达目标航点； 6. 机器人到达所选花盆附近后，调用浇水模块功能，对花盆浇水； 7. 浇水动作结束，用户界面切换回最初的等待界面，机器人转回等待模式。 | 是 |
| 4 | 自动巡检浇水 | 1. 用户启动机器人； 2. 用户界面切换至等待界面，提供“巡检自动浇水”候选模式； 3. 用户选择“巡检自动浇水”，机器人转入“巡检自动浇水”模式； 4. 机器人多次调用导航模块功能，自主规划路径依次到达“缺水花盆队列”中的各花盆处； 5. 机器人到达某个花盆附近后，调用浇水模块功能，对花盆浇水； 6. 浇水动作结束，机器人进而继续导航至下一个缺水花盆。 7. 待“缺水花盆队列”中的所有花盆都已浇水后，巡检自动浇水结束，用户界面切换回最初的等待界面，机器人转回等待模式。 | 是 |
| 5 | 巡检维护 | 1. 机器人巡检整个温室中的所有花盆，上报新增的花盆和被移动后原本位置缺失的花盆； 2. 更新植物数据库。 | 是 |
| 6 | 等待服务 | 1. 用户可在手动建图后设置机器人在温室中的初始位置； 2. 用户可以在等待服务模式下使用导航功能。 | 是 |

## 非功能需求

本项目系统的非功能需求需全面覆盖SRS-软件测试文档中[1]的“非功能需求”部分，非功能需求描述如下表所示。

**表6 非功能需求表**

|  |  |  |  |
| --- | --- | --- | --- |
| 需求编号 | 非功能需求 | 简要描述 | 是否测试 |
| 1 | 响应时间 | 用户界面能够在1秒内做出响应，在系统负载较大条件下保证响应时间不超过3秒。 | 是 |
| 2 | 功耗 | 系统设计尽可能强调节能特性，最大程度减少不必要的能耗。 | 是 |
| 3 | 异常处理能力 | 整个嵌入式系统都应具备应对异常情况的处理能力，该部分已在2.4中详细描述。 | 是 |
| 4 | 系统可用性 | 系统应提供简洁的用户操作方式，以及强大的自我检测与恢复机制。 | 是 |
| 5 | 可移植性 | 在仿真调试阶段，凡是支持ROS框架的平台均能顺利迁移并运行；在实验室实际测试阶段，不同型号的机器人只需适当配置，都能实现本项目的各项核心功能，包括前端交互和机器人系统功能。 | 否：目前缺乏可移植性测试设备 |

# 软件测试环境

## Gazebo仿真环境

Gazebo平台用于进行仿真测试。测试人员通过创建仿真环境、编写ROS节点，模拟机器人与仿真场景的复杂交互场景。相较于实验室硬件测试，仿真测试可快速验证开发成果，操作简便，一定程度缓解了实验设备不足的问题。

### 软件项

**表7 仿真测试软件项及其描述**

|  |  |
| --- | --- |
| **软件项** | **描述** |
| Ubuntu 18.04 LTS | 机载电脑的操作系统，支持ROS开发和运行 |
| ROS Melodic | 机器人操作系统 |
| Rviz | ROS机器人开发可视化工具 |
| Gazebo | ROS机器人仿真平台 |
| Python2 | 编写ROS运行脚本的解释环境 |
| Vue3 | 前端搭建与运行框架 |
| ROS-WebBridge | 支持前端页面和ROS层交互的框架 |
| SQLite | 适用于嵌入式开发的轻型数据库 |
| Visual Studio Code | 用于编辑调试的代码编辑器 |
| ROS驱动代码和功能包 | 支持ROS基础功能的正常运行 |

### 硬件及固件项

**表8 仿真测试硬件及固件项描述**

|  |  |
| --- | --- |
| **硬件及固件项** | **描述** |
| 个人电脑 | 用于代码开发、调试和仿真 |

### 其它材料

**表9 仿真测试其它材料描述**

|  |  |
| --- | --- |
| **其它材料项** | **描述** |
| 《ROS机器人开发实践》 | ROS开发指导书 |
| ROS Wiki | 提供ROS介绍、安装、教程、软件包的网站 |

### 参与组织

**表10 仿真测试人员组织表**

|  |  |  |
| --- | --- | --- |
| **组织名称** | **人员** | **主要职责** |
| 北航优质小组 | LTT | 项目组长、架构设计、前端开发 |
| Hao7un | 技术人员、测试人员、功能包开发 |
| Kun | 技术人员、测试人员、功能包开发 |
| Mxode | 技术人员、测试人员、功能包开发 |
| Le | 技术人员、测试人员、功能包开发 |

### 测试人员

项目组长LTT负责统筹测试规划，其余成员分别测试各自实现模块，并统一参与代码集成测试。

## 实验室真实环境

本项目的线下真实测试环境为新主楼G1027室。技术人员通过将开发代码部署在机器人上，对机器人进行功能验证和性能优化。“实验室线下测试”是开发过程中不可或缺的一环。

### 软件项

**表11 实验室测试软件项描述**

|  |  |
| --- | --- |
| **软件项** | **描述** |
| Ubuntu 18.04 LTS | 机载电脑的操作系统，支持ROS开发和运行 |
| ROS Melodic | 机器人操作系统 |
| Rviz | ROS机器人开发可视化工具 |
| Gazebo | ROS机器人仿真平台 |
| Python2 | 编写ROS运行脚本的解释环境 |
| Vue3 | 前端搭建与运行框架 |
| ROS-WebBridge | 支持前端页面和ROS层交互的框架 |
| SQLite | 适用于嵌入式开发的轻型数据库 |
| Visual Studio Code | 用于编辑调试的代码编辑器 |
| ROS驱动代码和功能包 | 支持ROS基础功能的正常运行 |

### 硬件及固件项

**表12 实验室测试硬件及固件项描述**

|  |  |
| --- | --- |
| **硬件及固件项** | **描述** |
| ROS机器人及配套机械臂 | 机器人产品的主体部分 |
| 开发板、继电器和水泵 | 控制浇花的下游套件 |
| 机载电脑 | 用于机上测试 |
| 个人电脑 | 用于开发和现场调试 |

### 其它材料

**表13 实验室测试其它材料**

|  |  |
| --- | --- |
| **其它材料项** | **描述** |
| 《ROS机器人开发实践》 | ROS开发指导书 |
| ROS Wiki | 提供ROS介绍、安装、教程、软件包的网站 |

### 参与组织

**表14 仿真测试人员组织表**

|  |  |  |
| --- | --- | --- |
| **组织名称** | **人员** | **主要职责** |
| 北航优质小组 | LTT | 项目组长、架构设计、前端开发 |
| Hao7un | 技术人员、测试人员、功能包开发 |
| Kun | 技术人员、测试人员、功能包开发 |
| Mxode | 技术人员、测试人员、功能包开发 |
| Le | 技术人员、测试人员、功能包开发 |

### 测试人员

项目组长LTT负责统筹测试规划，各组员均到现场参与各模块的测试工作，涵盖用户前端测试、基础模块测试、核心功能及例程测试等。

# 计划

## 总体设计

### 测试级别划分

**（1）单元测试**

**目标：**

确保项目中最小可测试单位（包括但不限于函数、类、模块）严格遵循设计规范，正确实现预定功能。特别是对于节点的基本通讯能力、传感器数据处理逻辑、基本算法实现等核心功能进行深度验证。

通过及时发现并修复底层错误，加速开发周期，使得开发团队能够更加自信地推进项目至更复杂的阶段。

**内容：**

i. 硬件单元测试：单独测试电机、水泵、继电器等硬件组件，确保它们能准确响应控制信号并提供预期输出，包括响应时间、负载能力、稳定性测试。并确认硬件与系统其他部分（如驱动程序、通信协议）间的无缝对接。

ii. 软件服务端单元测试：针对ROS中的各个服务节点进行测试，检查它们的输入输出逻辑是否符合设计要求。针对每个服务节点，模拟各种输入情景，验证其返回结果的一致性和准确性。故意触发异常情况，检验服务节点的错误处理机制是否有效。

iii. 用户界面单元测试：对UI组件进行单独测试，确保每个按钮、表单、数据显示等功能正确无误。

**方法：**

采用黑盒和白盒混合测试，对于ROS功能包进行黑盒测试，对于机器人以及前端交互界面更多采用白盒测试，直接检查内部逻辑、路径覆盖情况，确保每条代码路径都经过验证。。

**（2）集成测试**

**目标：**

在单元测试的基础上，验证不同模块或子系统之间的交互是否正常。重点检查数据流、控制信号传递的准确性以及各部分协同工作的能力。

**内容：**

逐步集成并测试，开始可以是小范围的几个模块组合，逐步扩大至整个系统。利用ROS的topic、service、action等通信机制进行交互测试。

**方法：**

采用黑盒测试方法，关注接口行为而不深入内部实现，同时也适当考虑内部逻辑。

**（3）系统测试**

**目标：**

验证整个系统是否满足业务需求和用户需求，包括功能完整度、性能、稳定性等方面。

**内容：**

进行功能测试、性能测试、稳定性测试、安全性测试等。确保机器人能在真实或仿真环境中完成温室浇花任务，如路径规划、精准浇水、环境监测与响应等。

**方法：**

多采用黑盒测试，模拟实际应用场景，检验系统端到端的功能表现。

**（4）迭代回归测试**

**目标：**

在每次功能更新或修复后，重新运行一套选定的测试用例，确保修改没有引入新的错误，同时原有功能依然稳定。

**内容：**

结合持续集成（CI），每次代码提交自动触发回归测试，快速反馈问题。重点在于选择覆盖核心功能和已知问题区域的测试用例，利用自动化工具提高效率。

### 测试类别划分

**（1）功能测试**

**i. 静态测试**

团队内会进行代码审查 (Code Review)，成员之间相互审查代码，查找潜在的编程错误、逻辑漏洞或不符合编码规范的地方。

此外，我们还会进行静态分析，使用自动化工具自动检测代码中的潜在缺陷，如未使用的变量、潜在的内存泄漏、不安全的代码模式等。

**ii. 白盒测试**

我们的白盒测试分为逻辑测试和单元测试。逻辑测试包括语句覆盖、分支覆盖、路径覆盖等，确保所有代码路径都经过测试。单元测试则是针对Core、Pending等功能模块编写测试用例，针对每个节点或关键函数进行独立测试，确保其功能正确。

我们确保测试用例能够执行到源代码中的每一条语句至少一次。实现此目标通常需要设计多个测试场景，每个场景专门针对一部分代码逻辑，通过不同的输入参数来触达所有可能的执行路径。

在此层次上，测试需确保程序中的每一个逻辑判断（即if语句、switch-case结构等）的真假分支都被至少测试一次。这要求设计测试案例时充分考虑所有逻辑分支的可能性，通过精确控制输入值来遍历这些分支。

**iii. 黑盒测试**

我们的黑盒测试分为场景测试和边界值测试。其中场景测试基于用户故事和使用场景设计测试案例，验证系统能否正确响应预期的输入并产生正确的输出；边界值测试则是选取边界条件（如数值的上下限附近值）进行测试，因为这些往往是错误高发区域。

**（2）性能测试**

**i. 负载测试**

模拟系统在低、中、高负载下的操作，评估系统的响应时间、吞吐量和资源使用情况。

**ii. 压力测试**

逐渐增加负载直至系统崩溃，确定系统的最大承受能力及崩溃点。

**iii. 稳定性测试**

长时间运行系统，检查是否有性能衰退、内存泄漏等问题。

**iv. 资源消耗测试**

监控系统运行时CPU、内存、硬盘等资源的使用情况，确保资源使用高效合理。

### 一般测试条件

**（1）环境匹配与配置一致性**

确保所有硬件设备（包括但不限于机器人本体、传感器、机载计算机、水泵）的型号、规格与软件版本相匹配，且与测试环境中的配置保持一致，减少因环境差异导致的测试偏差。

软件环境需包含ROS Melodic版本的稳定性和兼容性验证，以及所有依赖包的版本控制，避免因第三方库更新引起的不兼容问题。

**（2）人员要求与职责**

**测试执行者：**

至少2名团队成员参与，一名负责执行测试步骤，另一名负责记录测试结果和观察异常情况，确保测试的客观性和全面性。

**测试监督与评审：**

指定具有专业知识的高级成员（组长）定期审查测试过程和结果，确保测试的质量。

**（3）环境与资源管理**

**实验室环境标准：**

实验室应具备稳定的电源、网络连接、适宜的温湿度条件，并配备必要的安全设施，如紧急停机按钮、消防设备等。

**Gazebo仿真环境：**

当实验室不可或存在其他限制时，我们应充分利用Gazebo进行仿真测试。确保仿真模型与实物参数高度一致，包括物理属性、传感器精度、执行器延迟等。

**（4）进度与灵活性调整**

**灵活测试计划：**

测试计划应与开发进度紧密同步，若开发遇到延迟，测试计划可适当调整，但需确保关键里程碑测试不受影响。

**风险管理与应对：**

我们团队会识别可能影响测试进度的风险因素（如硬件故障、关键人员变动、技术难题），制定相应的风险缓解措施和备选方案。

**迭代测试策略：**

我们采取敏捷测试方法，允许在开发周期内多次迭代测试，通过GitLab的议题等快速反馈问题并调整，确保质量问题得到及时解决。

**（5）文档与沟通**

**测试文档：**

每次测试前我们应准备好详细的测试计划、测试用例、预期结果等文档，测试后编写测试报告，记录测试过程、结果、发现问题及改进建议。

**沟通机制：**

我们需要建立高效的沟通渠道，如定期的测试汇报线上、线下会议，确保测试信息及时传达给所有相关团队成员，促进问题的快速响应和解决。

### 测试过程

我们采用渐进式测试，强调逐步集成和验证系统组件，确保每一阶段都经过彻底测试后再进入下一阶段。

**（1）需求评审与文档准备**

我们会首先对SRS进行详细评审，确保所有参与者对项目目标、功能需求、非功能需求等有清晰共识。而后基于SRS文档，制定详尽的测试计划，包括测试策略、资源分配、时间表、风险评估及应对措施。

**（2）代码与设计评审**

我们分阶段进行代码静态审查，首先针对ROS部分的C++、Python代码，然后是系统服务端代码，最后是用户界面的Vue代码。我们使用代码审查工具辅助，确保代码质量、可读性及遵循最佳实践。

此外，我们对系统架构和重要模块的设计文档进行评审，确保设计满足需求且易于测试。

**（3）单元测试与模块验证**

我们分别进行硬件和软件单元测试。硬件单元测试中，我们对机器人各硬件组件进行独立功能和性能测试，记录测试数据，确保硬件的稳定性和可靠性；软件单元测试中，分层次进行单元测试，包括系统服务端的接口测试、用户界面功能测试，利用自动化测试框架编写测试用例，提高测试效率和覆盖率。

**（4）集成测试阶段**

我们选择增量式集成的方式，即，采用自底向上的方式逐步集成模块，先集成低层级模块，通过后再集成高层级模块。每次集成后进行测试，验证接口间通信和数据流的正确性。我们还需要确保集成后的系统能满足既定功能需求，执行场景测试、边界值测试等，验证系统功能的完整性。

在集成测试后期，我们需要对系统整体性能进行测试，包括响应时间、吞吐量、资源利用率等，确保系统在集成状态下仍能高效运行。

**（5）系统测试**

我们在真实或接近真实的环境下，对整个系统进行端到端的功能测试，验证系统是否满足业务流程和用户需求，并评估用户界面的易用性、直观性及满意度。

同时，我们还会进行长时间稳定性测试，确保系统在持续运行中的稳定表现。

**（6）回归测试与持续迭代**

每次新功能添加或缺陷修复后，我们均执行回归测试，确保现有功能未受影响，使用自动化工具加速回归测试过程。同时根据测试反馈和项目进度，灵活调整测试计划和优先级，确保关键问题优先解决，优化测试资源分配。

**（7）测试总结与反馈循环**

在每个测试阶段结束后，编写详细的测试报告，记录测试结果、发现的问题、问题解决情况及改进建议。同时利用反馈结果，促进代码和设计的迭代优化。我们定期组织复盘会议，总结测试过程中的经验教训，不断改进测试策略和方法。

### 测试结果记录规范

本节将主要阐述软件测试计划中标识的测试期间和测试之后要使用的数据记录、归纳和分析过程这些过程包括记录测试结果、将原始结果处理为适合评价的形式，以及保留数据归约与分析结果可能用到的手工、自动、半自动技术。

**（1）****测试项目整体结果记录规范**

为了确保测试过程的透明度和可追溯性，本节详细规定了测试项目整体结果的记录规范。此规范旨在为项目团队提供一个统一的标准，以便于评估测试活动的成效并有效管理测试数据。以下是具体要求和说明。

**表15 测试项目整体结果记录内容概览**

|  |  |
| --- | --- |
| 记录项 | 描述 |
| 测试项目标识 | 指测试项目的唯一识别代码或名称，用于区分不同的测试项目。应包括项目编号和简短描述，确保每个测试项目都能被准确无误地识别。 |
| 项目执行人 | 记录负责执行该测试项目的测试人员的全名。 |
| 测试时间 | 详细记录测试开始及结束的具体时间，格式为“YYYY-MM-DD HH”，即精确到小时。例如，“2024-05-05 14:00至2024-05-05 16:00”。 |
| 测试地点 | 明确指出测试执行的具体地理位置，包括实验室编号或其他任何有助于定位测试环境的信息。例如，“新主楼G座，1027实验室”。 |
| 整体评价 | 基于所有相关测试用例的执行结果，给出一个概括性的整体评价。评价应涵盖以下方面：   * 结果概览：总结测试用例的通过率、失败率及任何未执行的情况。 * 预期符合度：明确指出测试结果是否与预期相符，包括正面确认（如“所有关键功能均按预期执行”）或负面发现（如“发现X个异常情况，与预期行为不符”）。 * 问题概述：简要描述遇到的主要问题或缺陷，包括系统错误、硬件故障或测试流程中的不足。 * 改进建议：基于测试结果，提出对产品改进或后续测试策略的初步建议。 * 结论与后续行动：总结本次测试的关键发现，并概述下一步行动计划。 |

**（2）静态评审结果记录规范**

静态代码评审是确保软件质量的关键前期步骤，旨在识别潜在的编程错误、设计缺陷和不符合编码标准的情况，而不实际执行代码。以下是我们对静态评审结果记录的详细扩充规范，旨在提高评审效率和结果的可用性。

**表16 静态评审记录内容概览**

|  |  |
| --- | --- |
| 记录项 | 描述 |
| 测试项目标识 | 明确标注进行静态评审的测试项目的唯一代码或名称，便于跨文档追踪和关联。例如，“TST-WJ-01：水泵集成测试”。 |
| 项目静态评审员 | 记录执行静态代码评审的人员的全名。 |
| 代码风格 | * 层次结构：评价代码模块化程度，函数与类的组织是否合理，是否易于理解和维护。 * 命名约定：检查变量、函数、类的命名是否遵循项目命名规范，是否能够直观反映其用途。 * 注释质量：评估注释是否充分且不过度，是否清晰解释了复杂逻辑、算法或特定实现选择的原因。 * 代码格式：代码缩进、空白使用、行长度等是否符合团队约定的编码标准。 |
| 代码检查 | * 逻辑错误：记录发现的任何逻辑错误，包括但不限于无效的条件判断、不当的循环控制、资源泄露的潜在风险等。 * 性能考量：提供关于提高代码效率的建议，如算法优化、减少冗余计算、内存使用优化等。 * 兼容性与可移植性：评估代码在不同平台或环境下的兼容性，以及是否存在依赖特定环境的代码。 |
| 补充内容（可选） | * 最佳实践建议：提出遵循的最佳编程实践或设计模式，以提升代码质量和可维护性。 * 文档更新需求：指出代码相关的文档（如设计文档、用户手册）中需要同步更新的部分。 * 待讨论问题：记录评审过程中发现的不确定或需进一步讨论的问题，以备后续会议或工作坊讨论。 * 开发者反馈：预留空间记录代码作者对评审意见的反馈及后续修正计划，促进沟通和迭代。 |

**（3）单元测试用例执行结果记录总规范**

为确保软件质量及测试过程的系统性，所有单元测试用例执行后的记录需遵循以下综合规范，覆盖机器人功能测试与后端、前端接口测试的各个方面。这不仅有助于识别和解决问题，也促进了测试结果的透明度和可追溯性。

**表17 单元测试记录内容概览**

|  |  |
| --- | --- |
| 记录项 | 描述 |
| 测试用例标识 | 每个测试用例的唯一编号或名称，便于识别和索引。如“WJ-ROBOT-MV-01”。 |
| 被测单元名称 | 明确指出被测试的具体模块或功能单元，无论是机器人组件（如“机械臂控制单元”）、后端API（如“/map\_provider/auto\_init\_map”）还是前端页面（如“登录界面”）。 |
| 优先级 | 衡量测试用例的测试优先级程度，分别P0-P4，P0代表最高优先级。 |
| 测试输入 | 详述测试时向被测单元提供的具体输入数据。对于机器人测试，包括输入的命令类型（语音、图像等）；对于接口测试，则是具体的请求参数。 |
| 测试步骤 | 详述测试时向被测单元提供的具体操作指令。对于机器人测试，包括输入的命令具体内容；对于接口测试，则是具体的用户交互动作。 |
| 测试输出/行为 | * 机器人单元：记录从机器人控制系统接收到的所有反馈信息，如识别的语音文本、传感器读数、系统日志等。详细描述机器人响应测试输入后展现的实际物理行为，如移动路径、动作完成度等。 * 前后端接口：记录测试后获得的直接输出，如HTTP响应状态码、返回数据、页面渲染结果或数据库更改状态等。 |
| 测试成功标准/评价 | 综合评估测试结果是否达到预期，包括但不限于功能正确性、性能指标、异常处理能力及用户体验。需指出任何偏差、失败点或潜在风险，并评估这些因素对系统整体功能的影响。 |
| 补充内容（可选） | 记录任何额外的观察、分析、建议或需进一步探讨的议题，如测试环境的特殊配置、测试中发现的非目标问题、未来测试改进建议等。 |

**（4）集成测试用例执行结果记录规范**

集成测试作为软件开发过程中的重要阶段，旨在验证不同模块或子系统间协同工作的能力。为了确保集成测试的有效性及测试结果的可追溯性，制定详细的执行结果记录规范至关重要。

**表18 集成测试用例执行记录内容概览**

|  |  |
| --- | --- |
| 记录项 | 描述 |
| 测试用例标识 | 每个集成测试用例的唯一编号或名称，如“WJ-INT-Navigation-01”，以便于识别和后续跟踪。 |
| 优先级 | 衡量测试用例的测试优先级程度，分别P0-P4，P0代表最高优先级。 |
| 被测功能 | 明确描述测试用例旨在验证的特定功能或系统集成点，包括涉及的多个子系统或模块的交互。例如，“自动建图与导航的联动测试”。 |
| 测试输入 | 详尽记录执行测试时向系统提供的所有输入数据或外部触发条件。应确保记录的信息足够详细，以便于复现测试场景。 |
| 测试步骤 | 详述测试时向被测单元提供的具体操作指令、操作序列，包括但不限于用户界面操作（如按钮点击、表单提交）、系统配置变更等。 |
| 测试输出/行为 | * 前端表现：记录测试过程中前端展示的任何变化，包括但不限于页面元素更新、提示信息、错误消息、加载状态等。 * 后端响应：系统后台处理结果，如数据库状态变化、API响应内容、日志输出等。 * 机器人行为：集成测试特有的，详细描述机器人响应测试输入后的行为表现，如导航至指定位置、执行浇水动作等。 |
| 测试成功标准/评价 | * 功能符合度：评估系统功能是否按预期集成并正常工作，包括功能实现的准确性和完整性。 * 性能指标：如果适用，记录响应时间、处理速度、系统资源消耗等性能数据，评估系统在集成后的性能表现。 * 稳定性与健壮性：观察系统在测试期间的稳定性，记录任何异常终止、崩溃或错误处理情况。 * 交互连贯性：评估各子系统间的交互是否流畅，是否存在延迟或冲突。 |
| 补充内容（可选） | * 环境说明：测试执行时的软硬件环境配置、网络状况等，有助于分析测试结果是否受环境影响。 * 问题重现步骤：如果发现缺陷或异常，详细记录复现该问题的步骤，有助于快速定位和修复。 * 改进建议：基于测试结果，提出对系统集成、测试流程或测试用例本身的改进建议。 * 依赖关系确认：记录并验证测试用例执行前所需的依赖条件是否全部满足，以及这些依赖对测试结果的潜在影响。 |

**（5）系统测试用例执行结果记录规范**

系统测试作为软件生命周期中确保整体功能、性能及安全性的重要环节，其测试用例的执行结果记录必须详尽且结构化，以支撑决策和后续的系统优化。

**表19 系统测试用例执行记录内容概览**

|  |  |
| --- | --- |
| 记录项 | 描述 |
| 测试用例标识 | 每个系统测试用例的唯一编号或名称，遵循项目内部的命名规则，例如“WJ-SYS-AutoMap-01”。 |
| 优先级 | 衡量测试用例的测试优先级程度，分别P0-P4，P0代表最高优先级。 |
| 测试类别 | 明确指出测试用例所属的类别，可能包括但不限于：   * 功能测试：验证系统功能是否符合规格说明书的要求。 * 性能测试：评估系统在高负载下的表现，如响应时间、吞吐量等。 * 压力测试：测试系统在超出预期负载时的极限性能和稳定性。 * 恢复性测试：检查系统在异常或故障后能否正确恢复服务。 |
| 测试流程 | 详细记录测试执行的全过程，包括但不限于：   * 准备阶段：测试环境的搭建、数据准备、工具配置等。 * 执行步骤：逐条列出测试执行的具体步骤，包括输入数据、操作顺序、期望结果等。 * 监控指标：在测试过程中需要关注的关键性能指标或系统状态。 * 异常处理：测试中遇到的任何意外情况及处理方法。 |
| 测试结果 | * 功能符合度：逐一对比预期结果与实际结果，记录功能测试点的通过情况。 * 性能指标：提供性能测试的各项关键数值，如平均响应时间、吞吐量、资源利用率等。 * 异常与故障：测试过程中发现的系统异常、错误信息及故障恢复情况。 |
| 测试成功标准 | * 功能符合度：评估系统功能是否按预期集成并正常工作，包括功能实现的准确性和完整性。 * 性能指标：如果适用，记录响应时间、处理速度、系统资源消耗等性能数据，评估系统在集成后的性能表现。 * 稳定性与健壮性：观察系统在测试期间的稳定性，记录任何异常终止、崩溃或错误处理情况。 * 交互连贯性：评估各子系统间的交互是否流畅，是否存在延迟或冲突。 |
| 补充内容（可选） | * 环境配置：记录测试详细配置，包括软硬件版本、网络设置等。 * 测试工具与脚本：使用的测试工具、自动化脚本版本等。 * 资源消耗：测试中消耗的资源统计，如CPU、内存、存储等。 * 外部依赖：测试中涉及的第三方服务或系统状态。 * 未来测试建议：对未来测试策略或测试设计的建议。 |

**（6）问题用例及回归测试结果记录规范**

在软件开发与测试流程中，针对首次测试未通过的用例进行问题追踪、修复及回归测试是确保软件质量的关键环节。

**表20 问题用例及回归测试结果记录内容概览**

|  |  |
| --- | --- |
| **记录项** | **描述** |
| 测试项目标识 | 测试项目的唯一代码或名称，如“WJ-PROJ-MP-01”，便于跨文档追踪和归档。 |
| 出现问题的测试用例 | 标记首次未通过测试的具体用例的唯一标识。 |
| 具体问题描述 | 详细说明测试过程中观察到的问题现象，包括错误信息、异常行为或不符合预期的输出等，确保描述足够清晰，以便开发者快速理解问题所在。 |
| 影响 | * 功能影响：评估问题对软件核心功能的影响范围，是否影响基本操作或导致功能缺失。 * 性能影响：分析问题是否引起系统性能下降，如响应时间延长、资源占用异常等。 * 用户体验：考虑问题是否直接影响用户操作流程或界面展示，造成不良体验。 |
| 可行的解决方案 | * 问题原因：开发者分析的导致问题的根本原因，包括代码逻辑错误、配置不当、外部依赖问题等。 * 解决方案：提出的修复措施，包括代码修改、配置调整、依赖升级等详细步骤。 * 预期效果：修复后预期达到的功能正确性、性能恢复或用户体验改善的描述。 |
| 回归测试1 | * 测试环境：记录执行回归测试的具体环境配置，确保与初次测试环境一致或符合测试要求。 * 测试结果：明确标注测试用例执行后的状态，是否通过或仍存在相同/新问题。 |
| 分析1 | * 修复验证：记录采取的具体修复措施及验证过程，包括修改代码的版本、测试数据或步骤。 * 未解决原因（如适用）：若问题未解决，详细说明未能达到预期修复效果的原因分析。 * 后续策略：提出下一步的解决方案或调整的测试计划。 |
| 回归测试&分析k…… | 若回归测试仍然存在问题，重复进行回归测试和分析。对于仍未解决的问题，每一轮回归测试与分析应重复记录，直至问题完全解决。每次记录应包含新的测试环境、结果、分析及解决方案调整。 |

## 计划执行的测试

### 单元测试

#### 单元测试目标

* 机器人单元测试旨在验证自动浇花机器人系统中的各个模块是否按照设计要求正常运行。我们将针对每个关键模块执行测试，包括移动模块、手柄控制、传感器模块（如雷达、摄像头）、扬声器、机械臂和电源系统等。
* 前端交互界面单元测试的目标是确保所有用户界面元素都能够准确响应用户操作，并正确显示相关数据。这包括按钮、滑动条、数据展示窗口以及其他交互组件的功能性和视觉一致性。
* ROS端的单元测试旨在验证ROS层次中各个软件包的功能性和稳定性，确保它们能够在实时操作环境中正确执行指令。

#### 单元测试用例

* **机器人**单元测试用例情况
  + 对于**移动模块**，测试用例将涵盖不同运动方向、运动速度等方面，以确保机器人在各种场景下能够准确移，从而满足最基础的需求。
  + **手柄控制模块**的测试用例将验证手柄输入是否正确响应，并确保机器人能够根据用户指令进行相应操作。
  + **雷达**、**摄像头**等传感器模块的测试用例将验证其数据采集和处理功能，以及对机器人所处的环境变化的适应能力。
  + **机械臂模块**的测试用例将验证机械臂的各项功能，测试机械臂移动操作的准确性和稳定性，从而保障浇花行为的正常执行。
  + **电源系统模块**的测试用例将确保机器人在运作过程中的功耗情况。
* **前端交互界面**单元测试用例情况
  + **顶栏功能**：顶栏功能测试重点在于模式切换的响应性和反馈，确保在不同模式间切换时按钮状态和显示正确。
  + **花盆列表**：花盆列表功能测试验证了用户能够有效地查看和操作花盆相关信息，如切换视图和定位。
  + **控制面板**：控制面板测试集中在各种模式（如导航、位置设置和地图管理）下的操作响应和状态变更，确保功能按钮按预期工作。
  + **底栏**：底栏功能测试则关注于系统的连接状态和急停功能的可靠性，确保在必要时可以有效地控制系统状态。
* **ROS端**单元测试用例情况
  + **controller Package**：测试机器人主控的控制功能以及状态转移的正确性，从而保障控制逻辑的正确性。
  + **navigation Package**：验证路径规划和动态避障算法的有效性，确保机器人能够安全有效地导航至目标位置。
  + **map\_provider Package**：测试建图节点的基础建图功能，确保机器人能够接收到最新的地图数据。
  + **object\_detect Package**：验证花盆检测的准确率和实时性，确保机器人能够识别温室环境中的花盆。
  + **robot\_arm Package**：测试机械臂的操作精度和稳定性，确保其能够执行精细的操控任务。
  + **pot\_database Package**：测试数据库管理功能，包括数据的增删查改操作功能的正确性。

#### 单元测试计划

在机器人单元测试中，我们计划对机器人硬件层面的各个子模块实施独立的验证测试，其中涵盖了移动模块、操控手柄、雷达以及摄像头等各类传感器、扬声器、机械臂组件及电力供应系统的实际情况。此类检测将实验室场景中执行，确保每个模块均能在真实的应用场景中展现出其应有的效能。

针对前端界面，我们进行了深入的白盒单元测试，确保每一个代码逻辑的准确性和健壮性。开发者编写了详尽的测试脚本，直接针对源代码中的函数和模块进行验证。这包括但不限于对组件的状态管理、事件处理逻辑及数据绑定机制的细致检查。测试范围广泛覆盖了各种边界条件和异常场景，以确保用户界面在各种交互下的表现符合预期。测试在多浏览器和设备环境下执行，所有的问题将被记录并跟踪，确保了跨平台兼容性和一致性。

在ROS功能包层面，我们也同样执行了严格的白盒单元测试。针对每个功能模块，我们设计了单元测试案例，直接对ROS节点、服务和消息传递机制进行验证。这些测试不仅考虑了正常操作流程，还特别包含了对错误处理、异常处理及系统资源管理的检验。通过在接近真实的模拟环境中运行这些测试，我们确保了软件包在部署到实体机器人前能够稳定工作，且能有效应对各种运行时挑战。测试结果和日志都被详细记录，便于我们快速定位并解决问题，促进了迭代效率的提升。

### 集成测试

#### 集成测试目标

集成测试将验证ROS端、前端交互界面以及机器人等多个模块之间的接口和交互是否正确，以及整个系统是否能够正常协同工作。在集成测试部分中，我们将重点关注导航、动态避障等功能，一方面进一步测试单元测试中测试完成的各个模块能否进行协调工作，同时为后续系统测试奠定基础。

#### 集成测试用例

集成测试用例将主要覆盖建图、导航与动态避障、浇水以及启动服务四个方面的功能，以确保系统能够在单元测试的基础上进一步以提供高效、准确的功能，从而进一步满足浇花机器人的业务需求。集成测试用例不仅包括验证单个组件在实际工作流中的响应，还要确保模块与模块之间能够无缝对接，共同完成从环境感知到决策执行的全过程。

* **建图**功能：测试机器人在实验室的不同环境下的建图功能的正确性，能够适应多种复杂情况下的建图。
* **导航与动态避障**功能：测试机器人在不同障碍物环境下的导航功能，验证机器人导航功能的鲁棒性。
* **浇水**功能：测试机器人在不同花盆数量场景下能否正确执行浇水功能，确保机器人的浇水功能能够支撑业务需求。
* **启动服务**功能：测试机器人启动、异常启动以及重启等功能，保证机器人最基本的启动功能能够正常运作。

#### 集成测试计划

在完成单元测试后，集成测试会进一步对浇花机器人的各种基础功能进行集成测试。集成测试将根据系统的模块化设计进行，首先对核心功能进行集成测试，然后逐步增加其他功能模块。

### 系统测试

#### 系统测试目标

系统测试将验证整个温室浇花机器人系统是否符合用户需求和设计规格，以及是否能够稳定可靠地运行。我们将从需求分析出发，重点关注系统的功能和性能，以确保系统能够满足用户的各项需求。

#### 系统测试用例

系统测试用例将覆盖系统的各项功能和场景，包括自动建图、手动建图、巡检维护、定点浇水、自动巡检浇水、等待服务等。测试用例将模拟用户的实际操作，以验证系统在各种情况下的表现和稳定性。

* **自动建图**：验证系统是否能够自动扫描并建立完整的区域地图，生成完整的区域地图。
* **手动建图**：验证系统能否满足用户够通过手柄控制的方式来辅助或完全建立区域地图。
* **巡检维护**：验证系统能探测到新的花盆和缺失的花盆并上报；确保用户能够初始化花盆的含水量信息，以供机器人进行自动估算并进行缺水识别，并给用户提供自主操作植物数据的接口。
* **定点浇水**：确保系统能够在指定的位置准确浇水，满足使用者通过前端交互界面指定花盆让机器人去浇水的需求。
* **自动巡检浇水**：验证系统能否按照设定的路线自动巡航并完成浇水，满足使用者通过前端交互界面制定巡航路线，机器人可以沿着巡航路线运动并且自动浇水的功能。
* **等待服务**：验证系统在无任务执行时的表现和响应，进入等待使用者指令的系统状态。

#### 系统测试计划

系统测试将在模拟环境和真实环境下进行，以确保系统在不同环境下都能够正常运行。系统测试将在进行了单元测试与集成测试后，确保浇花机器人的基础功能均已满足要求后进行，在此基础上进一步测试浇花机器人的各项业务需求能否正常执行。

## 测试用例

### 单元测试：机器人

#### 移动模块单元测试

**表21 移动模块单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROBOT-MV |
| 相关需求 | 所有业务需求 |
| 测试涉及对象 | 机器人移动硬件、速度控制模块 |
| 前置条件 | 条件1：机器人电源开启，并且机器人状态无异常 |

**（1）黑盒测试用例1**

**表22 运动模块单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-MV-01 |
| 优先级 | P0 |
| 测试输入 | 运动模块脚本测试数据：   * cmd\_vel.linear.x : 0.5 m/s * cmd\_vel.linear.y : 0 m/s * cmd\_vel.angular.z : 0 rad/s |
| 测试步骤 | 步骤1:启动机器人电源  步骤2:启动运动模块测试脚本  步骤3:停止执行测试脚本，记录机器人实际运动情况 |
| 期望输出 | 终端输出/cmd\_vel话题信息,显示前进速度0.5 m/s |
| 测试成功标准 | 机器人应直线前进，无偏移，速度维持在0.5 m/s，停止时无滑行 |
| 用例备注 | 无 |

**（2）黑盒测试用例2**

**表23 运动模块单元测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-MV-02 |
| 优先级 | P0 |
| 测试输入 | 运动模块脚本测试数据：   * cmd\_vel.linear.x : 0 m/s * cmd\_vel.linear.y : 0 m/s * cmd\_vel.angular.z : 0.5 rad/s |
| 测试步骤 | 步骤1:启动机器人电源  步骤2:启动运动模块测试脚本  步骤3:停止执行测试脚本，记录机器人实际运动情况 |
| 期望输出 | 终端输出/cmd\_vel话题信息,显示旋转速度0.5 rad/s |
| 测试成功标准 | 机器人应在原地顺时针旋转，角速度恒定，停止时无余震 |
| 用例备注 | 无 |

**（3）黑盒测试用例3**

**表24 运动模块单元测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-MV-03 |
| 优先级 | P0 |
| 测试输入 | 运动模块脚本测试数据：   * cmd\_vel.linear.x : -0.3 m/s * cmd\_vel.linear.y : 0 m/s * cmd\_vel.angular.z : 0 rad/s |
| 测试步骤 | 步骤1:启动机器人电源  步骤2:启动运动模块测试脚本  步骤3:停止执行测试脚本，记录机器人实际运动情况 |
| 期望输出 | 终端输出/cmd\_vel话题信息,显示后退速度0.3 m/s |
| 测试成功标准 | 机器人应直线后退，速度恒定为-0.3 m/s，停止时无滑行 |
| 用例备注 | 无 |

**（4）黑盒测试用例4**

**表25 运动模块单元测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-MV-04 |
| 优先级 | P0 |
| 测试输入 | 运动模块脚本测试数据：   * cmd\_vel.linear.x : 0 m/s * cmd\_vel.linear.y : 0.5 m/s * cmd\_vel.angular.z : 0 rad/s |
| 测试步骤 | 步骤1:启动机器人电源  步骤2:启动运动模块测试脚本  步骤3:停止执行测试脚本，记录机器人实际运动情况 |
| 期望输出 | 终端输出/cmd\_vel话题信息，显示侧向移动速度0.5 m/s |
| 测试成功标准 | 机器人应侧向移动，速度恒定为0.5 m/s，移动平稳无摆动 |
| 用例备注 | 无 |

#### 手柄单元测试

**表26 手柄单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROBOT-JS |
| 相关需求 | 手动建图业务需求 |
| 测试涉及对象 | 机器人手柄 |
| 前置条件 | 条件1：机器人电源开启，并且机器人状态无异常  条件2：手柄与机器人连接稳定  条件3：手柄控制节点启动且正常工作 |

**（1）黑盒测试用例1**

**表27 手柄单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-JS-01 |
| 优先级 | P1 |
| 测试输入 | 测试输入1:手柄左摇杆向前推到最大值，并保持3秒  测试输入2:手柄左摇杆向后推到最大值，并保持3秒  测试输入3:手柄左摇杆向左推到最大值，并保持3秒  测试输入4:手柄左摇杆向右推到最大值，并保持3秒 |
| 测试步骤 | 步骤1: 连接手柄与机器人，同时确认手柄与机器人连接处稳定  步骤2:按照一定的时间间隔(例如5秒)，按照顺序执行输入操作  步骤3:观察机器人移动方向与终端输出 |
| 期望输出 | 终端输出机器人运动信息，同时运动信息与实际操作的运动过程保持一致 |
| 测试成功标准 | 机器人移动方向与移速正常，和用户预期一致；低延迟且无错误响应 |
| 用例备注 | 测试过程中需要保证环境安全，机器人周围没有无关人员 |

**（2）黑盒测试用例2**

**表28 手柄单元测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-JS-02 |
| 优先级 | P1 |
| 测试输入 | 测试输入1:手柄左摇杆向前推到最大值，并保持3秒  测试输入2: 手柄右摇杆向左推至机器人转向180度  测试输入3:手柄右摇杆向右推至机器人转向360度  测试输入4:手柄左摇杆向前推到最大值，并保持3秒 |
| 测试步骤 | 步骤1: 连接手柄与机器人，同时确认手柄与机器人连接处稳定  步骤2:按照一定的时间间隔(例如5秒)，按照顺序执行输入操作  步骤3:观察机器人移动方向与终端输出 |
| 期望输出 | 终端输出机器人运动信息，同时运动信息与实际操作的运动过程保持一致 |
| 测试成功标准 | 机器人移动方向与移速正常，和用户预期一致；低延迟且无错误响应 |
| 用例备注 | 测试过程中需要保证环境安全，机器人周围没有无关人员 |

#### 雷达单元测试

**表29 雷达单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROBOT-RADAR |
| 相关需求 | 导航和建图业务需求 |
| 测试涉及对象 | 机器人雷达单元 |
| 前置条件 | 条件1: 机器人电源开启，并且机器人状态无异常  条件2: 雷达单元启动正常 |

**（1）黑盒测试用例1**

**表30 雷达单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-RADAR-01 |
| 优先级 | P0 |
| 测试输入 | 无外部输入，机器人在静止状态下对实验室场地环境进行扫描 |
| 测试步骤 | 步骤1: 启动机器人并检查雷达单元是否正常  步骤2:雷达扫描周围环境稳定后，观察扫描结果 |
| 期望输出 | 雷达单元应能够准确地扫描并返回周围环境数据，在Rviz上显示附近的实验室环境 |
| 测试成功标准 | Rviz上输出与真实实验室环境的障碍物状态保持一致 |
| 用例备注 | 测试静止状态下雷达的感知精度和能力 |

**（2）黑盒测试用例2**

**表31 雷达单元测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-RADAR-02 |
| 优先级 | P0 |
| 测试输入 | 测试输入1:控制手柄使机器人面向障碍物运行 |
| 测试步骤 | 步骤1: 启动机器人并检查雷达单元是否正常  步骤2:执行测试输入，同时不断观察机器人机载电脑输出 |
| 期望输出 | 雷达单元应能够准确地扫描并返回前方障碍物的边界信息，在Rviz上正确显示出障碍物 |
| 测试成功标准 | Rviz上输出与真实实验室环境的障碍物状态保持一致 |
| 用例备注 | 测试在运动状态下雷达的感知精度 |

#### 摄像头单元测试

**表32 摄像头单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROBOT-CAMERA |
| 相关需求 | 花盆识别需求 |
| 测试涉及对象 | 机器人Kinect摄像头单元 |
| 前置条件 | 条件1: 摄像头单元已正确安装于机器人上  条件2: 摄像头单元电源连接并启动正常，XBOX图案能够正常发光显示 |

**（1）黑盒测试用例1**

**表33 摄像头单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-CAMERA-01 |
| 优先级 | P1 |
| 测试输入 | 无外部输入，机器人在室内环境自动运行 |
| 测试步骤 | 步骤1: 启动机器人并检查摄像头单元是否正常工作  步骤2：启动点云数据与摄像头数据捕捉程序  步骤3:移动机器人，摄像头分析环境 |
| 期望输出 | 摄像头应能够清晰录制并正确识别环境中的花盆 |
| 测试成功标准 | 摄像头视频清晰，能够准确返回实验室环境图片 |
| 用例备注 | 测试摄像头的基础识别功能 |

**（2）黑盒测试用例2**

**表34 摄像头单元测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-CAMERA-02 |
| 优先级 | P1 |
| 测试输入 | 机器人在不同光照条件下运行，从较暗的环境到明亮环境 |
| 测试步骤 | 步骤1: 启动机器人并检查摄像头单元是否正常工作  步骤2：启动点云数据与摄像头数据捕捉程序  步骤3:关闭室内灯光，拉上窗帘，保证室内环境阴暗  步骤4：移动机器人，摄像头分析环境  步骤5:打开室内灯光，使室内环境变回光亮  步骤6:移动机器人，摄像头分析环境 |
| 期望输出 | 摄像头能够自动调整曝光和焦点，保持图像质量 |
| 测试成功标准 | 图像在各种光照条件下依然清晰可见，没有过曝或过暗的情况 |
| 用例备注 | 测试摄像头在不同光照环境下的性能稳定性 |

#### 扬声器单元测试

**表35 扬声器单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROBOT-SPEAKER |
| 相关需求 | 浇水语音需求 |
| 测试涉及对象 | 机器人扬声器单元 |
| 前置条件 | 条件1: 扬声器单元已正确安装于机器人上  条件2: 机器人正常启动且扬声器模块正常工作 |

**（1）黑盒测试用例1**

**表36 扬声器单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-SPEAKER-01 |
| 优先级 | P2 |
| 测试输入 | 测试输入1:启动扬声器测试程序，程序中调用机器人语音模块输出“现在开始浇水”语音 |
| 测试步骤 | 步骤1: 确保扬声器连接正常并已启动  步骤2：执行测试输入，同时观察机器人的扬声器输出 |
| 期望输出 | 扬声器发出清晰的“现在开始浇水”的声音 |
| 测试成功标准 | 声音清晰可辨，音量适中，无明显杂音或失真 |
| 用例备注 | 测试扬声器的音质 |

#### 机械臂单元测试

**表37 机械臂单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROBOT-ARM |
| 相关需求 | 浇花功能需求 |
| 测试涉及对象 | 机器人机械臂单元 |
| 前置条件 | 条件1: 机械臂单元已正确安装于机器人上  条件2: 机器人正常启动且机械臂模块正常工作 |

**（1）黑盒测试用例1**

**表38 机械臂单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROBOT-ARM-01 |
| 优先级 | P1 |
| 测试输入 | 通过脚本设置机器人的机械臂运动参数：   * ctrl\_msg.position[0] (升降高度，单位：米)：0.5 * ctrl\_msg.velocity[0] (升降速度，单位：米/秒)：0.6 * ctrl\_msg.position[1] (手爪指尖距，单位：米)：0.15 * ctrl\_msg.velocity[1] (手爪开合角速度，单位：度/秒)：5 |
| 测试步骤 | 步骤1: 确保机器人开机并且机械臂正常工作  步骤2：执行机械臂测试脚本  步骤3: 观察机械臂运动状态，并与设置的机械臂运动参数进行比较 |
| 期望输出 | 终端可以正确显示预先设置好的机械臂运动参数 |
| 测试成功标准 | 机械臂按照预设的路径进行运动，同时最终可以收回到初始位置 |
| 用例备注 | 无 |

### 单元测试：前端交互界面

#### 顶栏功能单元测试

**表41 顶栏功能单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-FRONTEND-APPBAR |
| 相关需求 | 模式切换功能 |
| 测试涉及对象 | 前端, ROS 端 |
| 前置条件 | 1. 前端服务器工作正常 2. 开启 debug 模式, 使用 devtool 进行调试 3. 关闭 debug 模式, ROS 端开启 |

**（1）黑盒测试用例1**

**表42 顶栏功能单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-APPBAR-01 |
| 优先级 | P1 |
| 测试输入 | * 无 |
| 测试步骤 | 1. 使用 devtool 分别更改全局模式 2. 点击其他模式进行切换 |
| 期望输出 | 切换时所有按钮不可用, 目标模式按钮显示进度反馈 |
| 测试成功标准 | 多次切换均成功且互不影响 |
| 用例备注 | 前置条件 1, 2 |

**（2）黑盒测试用例2**

**表43 顶栏功能单元测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-APPBAR-02 |
| 优先级 | P1 |
| 测试输入 | * ROS Bridge |
| 测试步骤 | 1. 点击其他模式进行切换 |
| 期望输出 | 切换时所有按钮不可用, 目标模式按钮显示进度反馈 |
| 测试成功标准 | 多次切换均成功且互不影响 |
| 用例备注 | 前置条件 1, 3 |

#### 花盆列表单元测试

**表44 花盆列表单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-FRONTEND-POTLIST |
| 相关需求 | 花盆信息展示功能 |
| 测试涉及对象 | 前端, ROS 端 |
| 前置条件 | 1. 前端服务器工作正常 2. 开启 debug 模式, 使用 devtool 进行调试 3. 关闭 debug 模式, ROS 端开启 |

**（1）黑盒测试用例1**

**表45 花盆列表单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-POTLIST-01 |
| 优先级 | P1 |
| 测试输入 | * 无 |
| 测试步骤 | 1. 点击花盆 2. 分别切换, 禁用, 定位按钮 |
| 期望输出 | * 点击花盆会展开详细信息 * 点击切换能够在点云和图片之间切换 * 点击禁用能够在禁用和启用之间切换 * 点击定位将会在地图上展示花盆位置 |
| 测试成功标准 | 多次切换均成功且互不影响 |
| 用例备注 | 前置条件 1, 2 |

**（2）黑盒测试用例2**

**表46 花盆列表单元测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-POTLIST-02 |
| 优先级 | P1 |
| 测试输入 | * ROS Web Bridge |
| 测试步骤 | 1. 点击花盆 2. 分别切换, 禁用, 定位按钮 |
| 期望输出 | * 点击花盆会展开详细信息 * 点击切换能够在点云和图片之间切换 * 点击禁用能够在禁用和启用之阿金切换 * 点击定位将会在地图上展示花盆位置 |
| 测试成功标准 | 多次切换均成功且互不影响 |
| 用例备注 | 前置条件 1, 3 |

#### 3D地图单元测试

**表47 3D地图单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-FRONTEND-MAP |
| 相关需求 | 地图展示功能 |
| 测试涉及对象 | 前端, ROS 端 |
| 前置条件 | 1. 前端服务器工作正常 2. 开启 debug 模式, 使用 devtool 进行调试 3. 关闭 debug 模式, ROS 端开启 |

**（1）黑盒测试用例1**

**表48 3D地图单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-MAP-01 |
| 优先级 | P1 |
| 测试输入 | * 无 |
| 测试步骤 | 1. 使用左键, 滚轮, 右键控制地图 |
| 期望输出 | * 左键移动 * 滚轮缩放 * 右键旋转 |
| 测试成功标准 | 地图可以正常移动 |
| 用例备注 | 前置条件 1, 2 |

**（2）黑盒测试用例2**

**表49 3D地图单元测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-MAP-02 |
| 优先级 | P1 |
| 测试输入 | * ROS Web Bridge |
| 测试步骤 | 1. 使用左键, 滚轮, 右键控制地图 |
| 期望输出 | * 左键移动 * 滚轮缩放 * 右键旋转 |
| 测试成功标准 | 地图可以正常移动 |
| 用例备注 | 前置条件 1, 3 |

**（3）黑盒测试用例3**

**表50 3D地图单元测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-MAP-03 |
| 优先级 | P1 |
| 测试输入 | * 无 |
| 测试步骤 | 1. 切换到跟随机器人模式, 待视角稳定后, 使用左键拖拽地图(多次, 间隔不超过 5s), 随后等待 5 s 2. 切换回地图模式, 使用左键拖拽地图, 等待 5s |
| 期望输出 | * 视角将平滑移动到机器人前方, 拖动后, 5s 内没有其他操作, 视角自动回到机器人前方 * 切换为地图模式后, 不会再跟随机器人 |
| 测试成功标准 | 模式正常切换, 不保留状态 |
| 用例备注 | 前置条件 1, 2 |

**（4）黑盒测试用例4**

**表51 3D地图单元测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-MAP-04 |
| 优先级 | P1 |
| 测试输入 | * ROS Web Bridge |
| 测试步骤 | 1. 切换到跟随机器人模式, 待视角稳定后, 使用左键拖拽地图(多次, 间隔不超过 5s), 随后等待 5 s 2. 切换回地图模式, 使用左键拖拽地图, 等待 5s |
| 期望输出 | * 视角将平滑移动到机器人前方, 拖动后, 5s 内没有其他操作, 视角自动回到机器人前方 * 切换为地图模式后, 不会再跟随机器人 |
| 测试成功标准 | 模式正常切换, 不保留状态 |
| 用例备注 | 前置条件 1, 3 |

**（5）黑盒测试用例5**

**表52 3D地图单元测试黑盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-MAP-05 |
| 优先级 | P1 |
| 测试输入 | * 无 |
| 测试步骤 | 1. 开启花盆定位模式, 移动视角到出现花盆, 点击花盆 2. 移动地图, 等待 5s |
| 期望输出 | * 视角围绕点击的花盆旋转, 花盆列表滚动到指定花盆并展开 * 移动地图后回到地图模式, 不会再次围绕花盆旋转 |
| 测试成功标准 | 模式正常切换, 不保留状态 |
| 用例备注 | 前置条件 1, 2 |

**（6）黑盒测试用例6**

**表53 3D地图单元测试黑盒测试用例6说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-MAP-06 |
| 优先级 | P1 |
| 测试输入 | * ROS Web Bridge |
| 测试步骤 | 1. 开启花盆定位模式, 移动视角到出现花盆, 点击花盆 2. 移动地图, 等待 5s |
| 期望输出 | * 视角围绕点击的花盆旋转, 花盆列表滚动到指定花盆并展开 * 移动地图后回到地图模式, 不会再次围绕花盆旋转 |
| 测试成功标准 | 模式正常切换, 不保留状态 |
| 用例备注 | 前置条件 1, 3 |

**（7）黑盒测试用例7**

**表54 3D地图单元测试黑盒测试用例7说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-MAP-07 |
| 优先级 | P1 |
| 测试输入 | * ROS Web Bridge |
| 测试步骤 | 1. 等待地图加载完全 2. 重启ROS端并移动机器人 |
| 期望输出 | * 恢复后地图与机器人的位置与实际保持一致 * 回复后地图与激光雷达的障碍现实一致 |
| 测试成功标准 | 地图同步功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

#### 控制面板单元测试

**表55 控制面板元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-FRONTEND-OPPANEL |
| 相关需求 | 特定模式相关功能 |
| 测试涉及对象 | 前端, ROS 端 |
| 前置条件 | 1. 前端服务器工作正常 2. 开启 debug 模式, 使用 devtool 进行调试 3. 关闭 debug 模式, ROS 端开启 |

**（1）黑盒测试用例1**

**表56 控制面板单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-01 |
| 优先级 | P0 |
| 测试输入 | * 处于等待(Pending)模式 |
| 测试步骤 | 1. 点击导航, 点击取消, 点击导航 2. 在地图上按下左键, 指定目的地, 鼠标移动指定目的朝向, 松开左键 3. 等待导航完成后重复 1, 2, 在导航未完成时点击取消 |
| 期望输出 | * 点击导航后,其他按钮不可用, 在未设置航点前可以取消导航, 其他按钮功能恢复 * 指定目的地和朝向后, 显示导航路线, 机器人移动到目标, 面板按钮重新可用 * 再次指定目的地和朝向后, 在导航过程中取消, 除返回结果不同外, 前端展示与上条相同 |
| 测试成功标准 | 导航功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（2）黑盒测试用例2**

**表57 控制面板单元测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-02 |
| 优先级 | P0 |
| 测试输入 | * 处于等待(Pending)模式 |
| 测试步骤 | 1. 点击手动设置位置, 点击取消, 点击手动设置位置 2. 在地图上按下左键, 指定位置, 鼠标移动指定朝向, 松开左键 |
| 期望输出 | * 点击手动设置位置后,其他按钮不可用, 在未设置前可以取消, 其他按钮功能恢复 * 指定位置和朝向后, 设置机器人到指定位置和朝向, 面板按钮重新可用 |
| 测试成功标准 | 手动设置位置功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（3）黑盒测试用例3**

**表58 控制面板单元测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-03 |
| 优先级 | P0 |
| 测试输入 | * 处于等待(Pending)模式 |
| 测试步骤 | 1. 点击自动设置位置 |
| 期望输出 | * 点击自动设置位置后,其他按钮不可用, 待设置完成后, 按钮功能恢复 |
| 测试成功标准 | 自动动设置位置功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（4）黑盒测试用例4**

**表59 控制面板单元测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-04 |
| 优先级 | P0 |
| 测试输入 | * 处于等待(Pending)模式 |
| 测试步骤 | 1. 点击保存地图 |
| 期望输出 | * 点击保存地图后,其他按钮不可用, 待保存完成后, 按钮功能恢复 |
| 测试成功标准 | 保存地图功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（5）黑盒测试用例5**

**表60 控制面板单元测试黑盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-05 |
| 优先级 | P0 |
| 测试输入 | * 处于等待(Pending)模式 |
| 测试步骤 | 1. 点击清除地图 |
| 期望输出 | * 点击清除地图后,其他按钮不可用, 待保存完成后, 按钮功能恢复 |
| 测试成功标准 | 清除地图功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（6）黑盒测试用例6**

**表61 控制面板单元测试黑盒测试用例6说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-06 |
| 优先级 | P0 |
| 测试输入 | * 处于自动建图(Auto\_Map)模式 |
| 测试步骤 | 1. 点击开始建图 2. 点击开始建图, 一段时间后点击取消 |
| 期望输出 | * 点击开始建图后, 机器人自动进行建图, 运行时显示反馈, 完成后显示结果 * 点击开始建图后, 机器人自动进行建图, 运行时显示反馈, 取消时显示结果 |
| 测试成功标准 | 自动建图功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（7）黑盒测试用例7**

**表62 控制面板单元测试黑盒测试用例7说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-07 |
| 优先级 | P0 |
| 测试输入 | * 处于手动建图(Manual\_Map)模式 |
| 测试步骤 | 1. 点击开始建图, 一段时间后点击结束建图 |
| 期望输出 | * 点击开始建图后, 手动进行建图, 运行时显示反馈, 完成后显示结果 |
| 测试成功标准 | 手动建图功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（8）黑盒测试用例8**

**表63 控制面板单元测试黑盒测试用例8说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-08 |
| 优先级 | P0 |
| 测试输入 | * 处于巡检维护(Maintain)模式 |
| 测试步骤 | 1. 点击开始巡检 2. 点击开始巡检, 一段时间后点击取消 |
| 期望输出 | * 点击开始巡检后, 机器人自动开始巡检, 运行时显示反馈, 完成后显示结果 * 点击开始巡检后, 机器人自动开始巡检, 运行时显示反馈, 取消后显示结果 |
| 测试成功标准 | 巡检维护功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（9）黑盒测试用例9**

**表64 控制面板单元测试黑盒测试用例9说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-09 |
| 优先级 | P0 |
| 测试输入 | * 处于定点浇水(Target)模式 |
| 测试步骤 | 1. 在花盆列表选择花盆 2. 点击开始 3. 点击开始, 一段时间后点击取消 |
| 期望输出 | * 点击开始后, 机器人自动开始对指定花盆浇水, 运行时显示反馈, 完成后显示结果 * 点击开始后, 机器人自动开始对指定花盆浇水, 运行时显示反馈, 取消后显示结果 |
| 测试成功标准 | 定点浇水功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（10）黑盒测试用例10**

**表65 控制面板单元测试黑盒测试用例10说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-OPPANEL-10 |
| 优先级 | P0 |
| 测试输入 | * 处于自动巡检浇水(Auto\_Water)模式 |
| 测试步骤 | 1. 点击开始 2. 点击开始, 一段时间后点击取消 |
| 期望输出 | * 点击开始后, 机器人自动开始对启用花盆浇水, 运行时显示反馈, 完成后显示结果 * 点击开始后, 机器人自动开始对启用花盆浇水, 运行时显示反馈, 取消后显示结果 |
| 测试成功标准 | 自动巡检浇水功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

#### 底栏单元测试

**表66 底栏元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-FRONTEND-FOOTER |
| 相关需求 | 连接状态, 急停功能 |
| 测试涉及对象 | 前端, ROS 端 |
| 前置条件 | 1. 前端服务器工作正常 2. 开启 debug 模式, 使用 devtool 进行调试 3. 关闭 debug 模式, ROS 端开启 |

**（1）黑盒测试用例1**

**表67 底栏单元测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-FOOTER-01 |
| 优先级 | P2 |
| 测试输入 | * ROS Web Bridge |
| 测试步骤 | 1. 保证 ROS 端工作正常, 开启前端 2. 关闭 ROS 端 3. 在 25 s 内重启 ROS 端 4. 关闭 ROS 端, 等待超过 25s 5. 重启 ROS 端完成后, 点击右侧连接按钮(此时应该显示 ERROR) |
| 期望输出 | * 前端连接到 ROS 端, 显示 CONNECTED * ROS 掉线后, 每 5s 尝试重连 * 重启 ROS 后, 正常连接 * 前端最多尝试重连 5 次, 5次未连接后显示 ERROR * 点击按钮后, 重新连接到 ROS 显示 CONNECTED |
| 测试成功标准 | 重连功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

**（2）黑盒测试用例2**

**表68底栏单元测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-FRONTEND-FOOTER-02 |
| 优先级 | P2 |
| 测试输入 | * ROS Web Bridge |
| 测试步骤 | 1. 使用导航功能使机器人移动, 点击紧急停止按钮 2. 一段时间后点击恢复 |
| 期望输出 | * 点击急停后, 机器人不再运动 * 点击恢复后, 机器人恢复运动 |
| 测试成功标准 | 急停功能工作正常 |
| 用例备注 | 前置条件 1, 3 |

### 单元测试：ROS端测试

#### controller Package

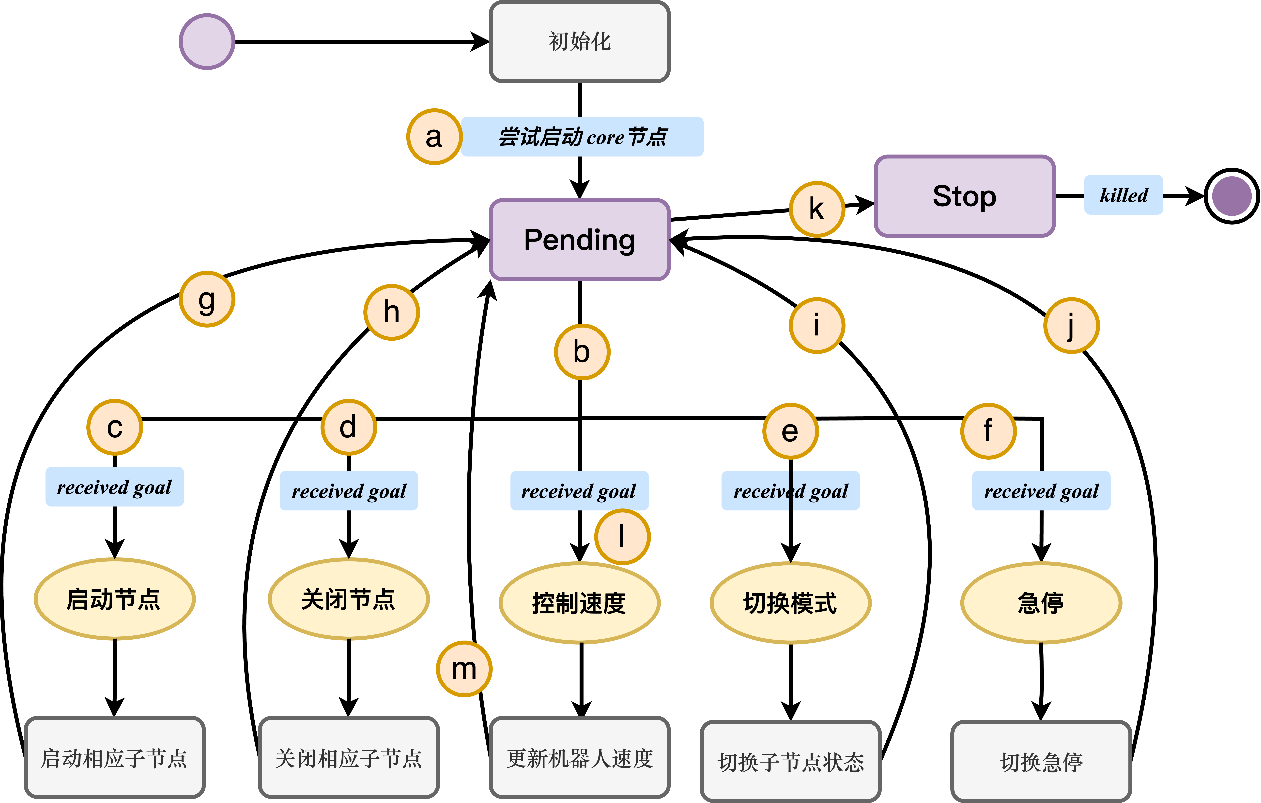


图 4.1 core节点白盒测试状态转移图

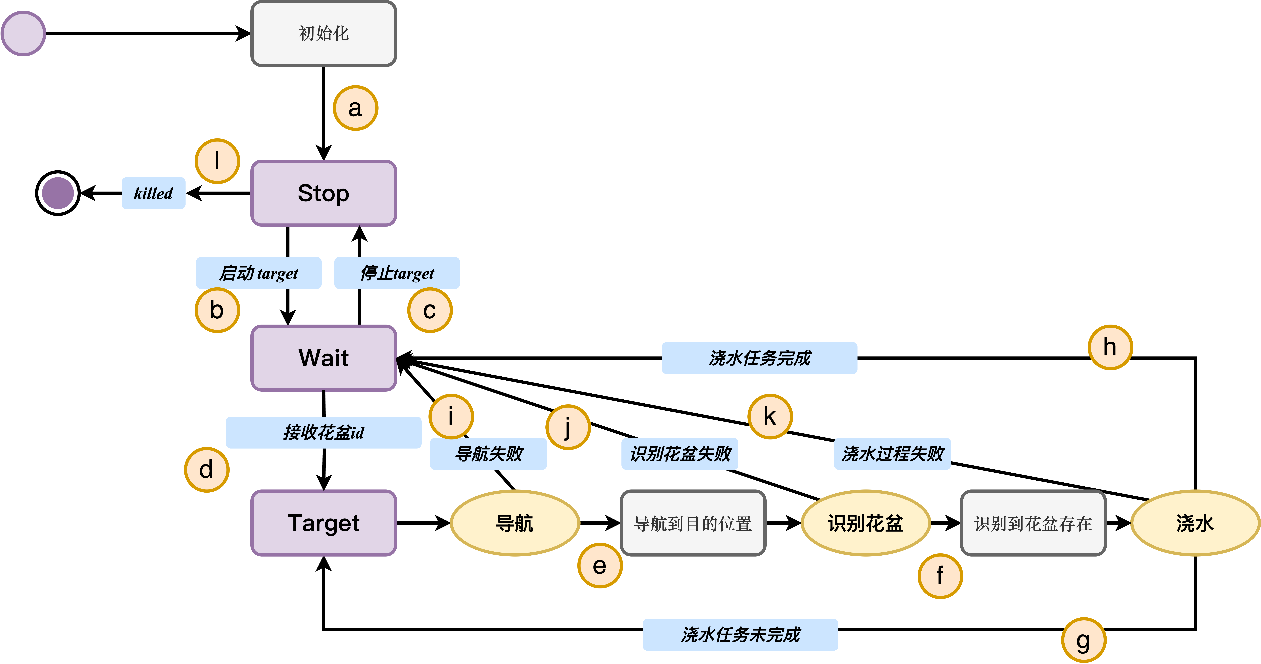


图 4.2 target节点白盒测试状态转移图

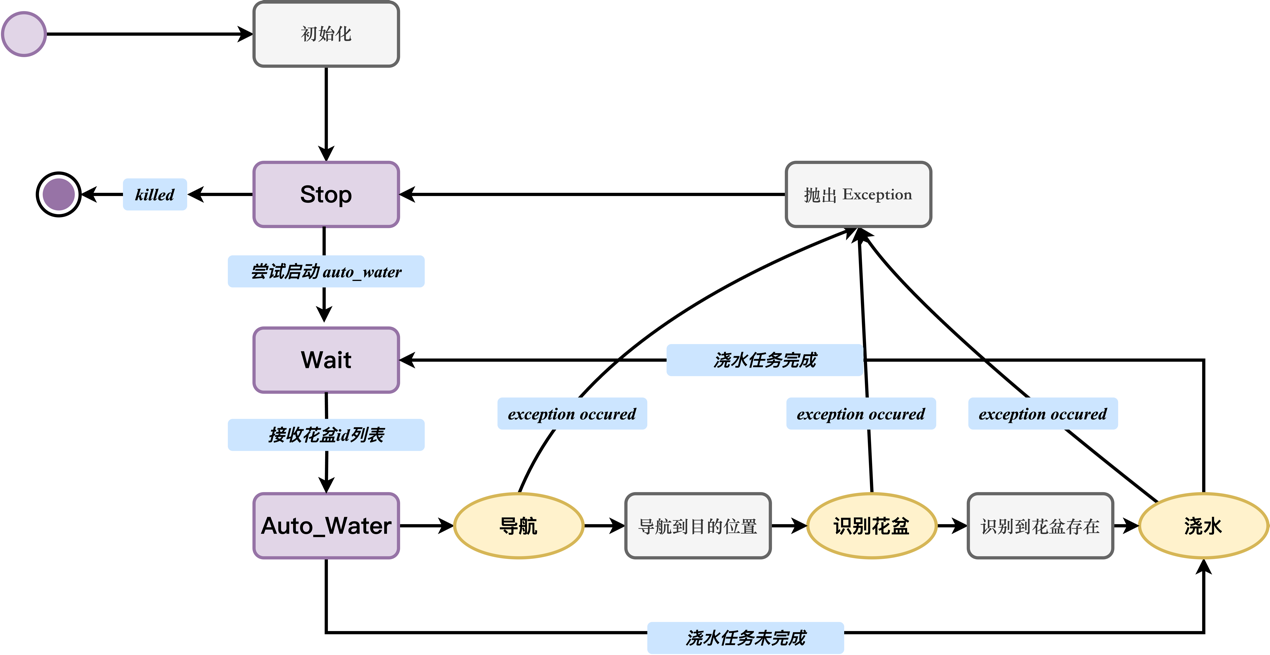


图 4.3 auto\_water白盒测试状态转移图

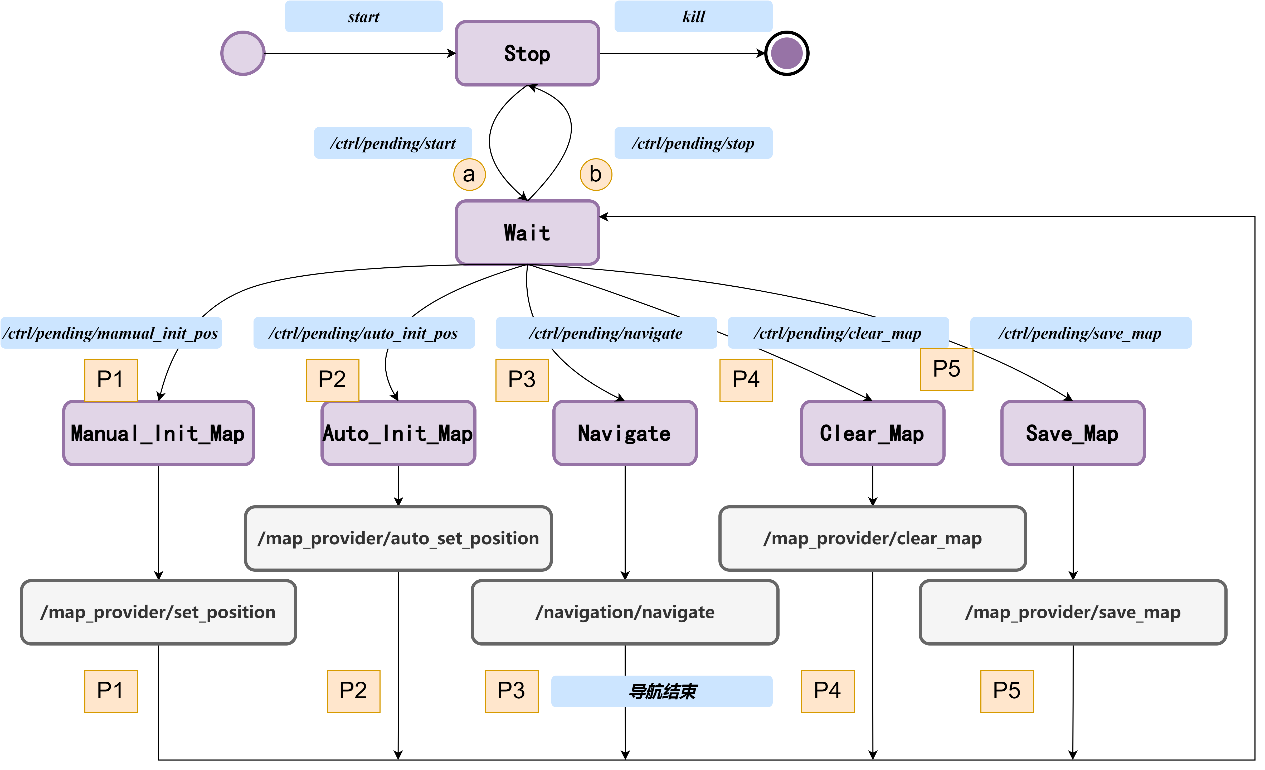


图 4.4 pending白盒测试状态转移图

**（1）白盒测试用例1**

**表69 controller包target节点单元测试白盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-TARGET-01 |
| 被测单元 | handle\_start |
| 优先级 | P1 |
| 测试输入 | 服务请求 |
| 测试步骤 | * 初始化ROS节点 * 调用handle\_start方法 |
| 期望输出 | * 输出日志信息 * 返回StartResponse |
| 测试成功标准 | * 当状态为STOP时，返回True，并且状态切换为WAIT； * 当状态不为STOP时，返回False，且不改变状态 |
| 用例备注 | 无 |

**（2）白盒测试用例2**

**表70 controller包target节点单元测试白盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-TARGET-02 |
| 被测单元 | handle\_stop |
| 优先级 | P1 |
| 测试输入 | 服务请求 |
| 测试步骤 | * 初始化ROS节点 * 调用handle\_stop方法 |
| 期望输出 | * 输出日志信息 * 返回StopResponse |
| 测试成功标准 | * 当状态为TARGET时，设置任务结果为'cancel'并返回True； * 当状态为WAIT时，若任务活动，则设置任务为Preempted并返回True；否则返回False |
| 用例备注 | 无 |

**（3）白盒测试用例3**

**表71 controller包target节点单元测试白盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-TARGET-03 |
| 被测单元 | hello\_callback |
| 优先级 | P2 |
| 测试输入 | 发布hello消息 |
| 测试步骤 | * 初始化ROS节点 * 发布Hello消息 |
| 期望输出 | * 接收Hello消息 * 发布NodeInfo消息 |
| 测试成功标准 | 接收Hello消息后发布NodeInfo消息，其中mode为5，state为当前状态 |
| 用例备注 | 无 |

**（4）白盒测试用例4**

**表71 controller包target节点单元测试白盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-TARGET-04 |
| 被测单元 | get\_target\_pose |
| 优先级 | P1 |
| 测试输入 | 目标ID |
| 测试步骤 | * 初始化ROS节点 * 调用get\_target\_pose方法并传入目标ID |
| 期望输出 | 返回目标位置 |
| 测试成功标准 | 成功获取目标位置信息并返回；若获取失败，记录警告日志并返回None |
| 用例备注 | 无 |

**（5）白盒测试用例5**

**表71 controller包target节点单元测试白盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-TARGET-05 |
| 被测单元 | navigate\_to\_target |
| 优先级 | P1 |
| 测试输入 | 目标位置信息 |
| 测试步骤 | * . 初始化ROS节点 * 调用navigate\_to\_target方法并传入目标位置信息 |
| 期望输出 | 导航至目标位置  返回导航结果 |
| 测试成功标准 | * 成功导航至目标位置并返回True； * 若导航失败，记录警告日志并返回False |
| 用例备注 | 无 |

**（6）白盒测试用例6**

**表71 controller包target节点单元测试白盒测试用例6说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-TARGET-06 |
| 被测单元 | check\_flowerpot |
| 优先级 | P2 |
| 测试输入 | 花盆ID |
| 测试步骤 | * 初始化ROS节点 * 调用check\_flowerpot方法并传入花盆ID |
| 期望输出 | * 检查花盆状态 * 返回检查结果 |
| 测试成功标准 | * 成功检查花盆状态并返回True； * 若检查失败，记录警告日志并返回False |
| 用例备注 | 无 |

**（7）白盒测试用例7**

**表71 controller包target节点单元测试白盒测试用例7说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-TARGET-07 |
| 被测单元 | execute\_cb |
| 优先级 | P0 |
| 测试输入 | 目标列表 |
| 测试步骤 | * 初始化ROS节点 * 调用execute\_cb方法并传入目标列表 |
| 期望输出 | * 执行浇水任务 * 返回任务执行结果 |
| 测试成功标准 | * 成功完成所有目标的浇水任务并返回成功； * 若导航、花盆检查或浇水任务失败，设置任务结果为失败并记录相应日志 |
| 用例备注 | 无 |

**（8）白盒测试用例8**

**表69 controller包auto\_water节点单元测试白盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-AUTO-01 |
| 被测单元 | handle\_start |
| 优先级 | P1 |
| 测试输入 | 服务请求 |
| 测试步骤 | * 初始化ROS节点 * 调用handle\_start方法 |
| 期望输出 | * 输出日志信息 * 返回StartResponse |
| 测试成功标准 | * 当状态为STOP时，返回True，并且状态切换为WAIT； * 当状态不为STOP时，返回False，且不改变状态 |
| 用例备注 | 无 |

**（9）白盒测试用例9**

**表70 controller包auto\_water节点单元测试白盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-AUTO-02 |
| 被测单元 | handle\_stop |
| 优先级 | P1 |
| 测试输入 | 服务请求 |
| 测试步骤 | * 初始化ROS节点 * 调用handle\_stop方法 |
| 期望输出 | * 输出日志信息 * 返回StopResponse |
| 测试成功标准 | * 当状态为TARGET时，设置任务结果为'cancel'并返回True； * 当状态为WAIT时，若任务活动，则设置任务为Preempted并返回True；否则返回False |
| 用例备注 | 无 |

**（10）白盒测试用例10**

**表71 controller包auto\_water节点单元测试白盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-AUTO-03 |
| 被测单元 | hello\_callback |
| 优先级 | P2 |
| 测试输入 | 发布hello消息 |
| 测试步骤 | * 初始化ROS节点 * 发布Hello消息 |
| 期望输出 | * 接收Hello消息 * 发布NodeInfo消息 |
| 测试成功标准 | 接收Hello消息后发布NodeInfo消息，其中mode为5，state为当前状态 |
| 用例备注 | 无 |

**（11）白盒测试用例11**

**表71 controller包auto\_water节点单元测试白盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-AUTO-04 |
| 被测单元 | navigate\_to\_target |
| 优先级 | P1 |
| 测试输入 | 目标位置信息 |
| 测试步骤 | * . 初始化ROS节点 * 调用navigate\_to\_target方法并传入目标位置信息 |
| 期望输出 | * 导航至目标位置 * 返回导航结果 |
| 测试成功标准 | * 成功导航至目标位置并返回True； * 若导航失败，记录警告日志并返回False |
| 用例备注 | 无 |

**（12）白盒测试用例12**

**表71 controller包auto\_water节点单元测试白盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-TARGET-05 |
| 被测单元 | check\_flowerpot |
| 优先级 | P2 |
| 测试输入 | 花盆ID |
| 测试步骤 | * 初始化ROS节点 * 调用check\_flowerpot方法并传入花盆ID |
| 期望输出 | * 检查花盆状态 * 返回检查结果 |
| 测试成功标准 | * 成功检查花盆状态并返回True； * 若检查失败，记录警告日志并返回False |
| 用例备注 | 无 |

**（13）白盒测试用例13**

**表71 controller包auto\_water节点单元测试白盒测试用例6说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-TARGET-06 |
| 被测单元 | execute\_cb |
| 优先级 | P0 |
| 测试输入 | 目标列表 |
| 测试步骤 | * 初始化ROS节点 * 调用execute\_cb方法并传入目标列表 |
| 期望输出 | * 执行浇水任务 * 返回任务执行结果 |
| 测试成功标准 | * 成功完成所有目标的浇水任务并返回成功； * 若导航、花盆检查或浇水任务失败，设置任务结果为失败并记录相应日志 |
| 用例备注 | 无 |

**（14）白盒测试用例14**

**表71 controller包pending节点单元测试白盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-PEND-01 |
| 被测单元 | start\_serve |
| 优先级 | P0 |
| 测试输入 | 服务请求 |
| 测试步骤 | * 初始化ROS节点 * 调用start\_serve方法 |
| 期望输出 | * 返回StartResponse(success=True) * 节点状态self.node\_info.state为Wait * 发布NodeInfo消息，state为Wait |
| 测试成功标准 | 当状态切换至Wait并且成功返回StartResponse(success=True)。 |
| 用例备注 | 无 |

**（15）白盒测试用例15**

**表71 controller包pending节点单元测试白盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-PEND-02 |
| 被测单元 | stop\_serve |
| 优先级 | P0 |
| 测试输入 | 服务请求 |
| 测试步骤 | * 初始化ROS节点 * 调用stop\_serve方法 |
| 期望输出 | * 返回StopResponse(success=True) * 节点状态self.node\_info.state为Stop * 发布NodeInfo消息，state为Stop |
| 测试成功标准 | 当状态切换至Stop并且成功返回StopResponse(success=True) |
| 用例备注 | 无 |

**（16）白盒测试用例16**

**表71 controller包pending节点单元测试白盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-PEND-03 |
| 被测单元 | clear\_map\_serve |
| 优先级 | P0 |
| 测试输入 | ClearMapRequest |
| 测试步骤 | * 初始化ROS节点 * 设置节点状态为Wait * 调用clear\_map\_serve方法 |
| 期望输出 | * 返回ClearMapResponse(success=True) * 节点状态self.node\_info.state切换至Clear\_Map并最终恢复到Wait * 发布ClearMap消息 |
| 测试成功标准 | 成功发布ClearMap消息并返回ClearMapResponse(success=True) |
| 用例备注 | 无 |

**（17）白盒测试用例17**

**表71 controller包pending节点单元测试白盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-PEND-04 |
| 被测单元 | auto\_init\_pos\_serve |
| 优先级 | P0 |
| 测试输入 | AutoInitPosRequest |
| 测试步骤 | * 初始化ROS节点 * 设置节点状态为Wait * 调用auto\_init\_pos\_serve方法 |
| 期望输出 | * 返回AutoInitPosResponse(success=True) * 节点状态self.node\_info.state切换至Auto\_Init\_Map并最终恢复到Wait * 输出日志信息"自动设置了机器人初始位置" |
| 测试成功标准 | 成功输出日志并返回AutoInitPosResponse(success=True) |
| 用例备注 | 无 |

**（18）白盒测试用例18**

**表71 controller包pending节点单元测试白盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-PEND-05 |
| 被测单元 | manual\_init\_pos\_serve |
| 优先级 | P0 |
| 测试输入 | ManualInitPosRequest |
| 测试步骤 | * 初始化ROS节点 * 设置节点状态为Wait * 调用manual\_init\_pos\_serve方法 |
| 期望输出 | * 返回ManualInitPosResponse(success=True) * 节点状态self.node\_info.state切换至Manual\_Init\_Map并最终恢复到Wait * 发布SetPosition消息 |
| 测试成功标准 | 成功发布SetPosition消息并返回ManualInitPosResponse(success=True) |
| 用例备注 | 无 |

**（19）白盒测试用例19**

**表71 controller包pending节点单元测试白盒测试用例6说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-PEND-06 |
| 被测单元 | * 初始化ROS节点 * 设置节点状态为Wait * 调用save\_map\_serve方法 |
| 优先级 | P0 |
| 测试输入 | SaveMapRequest |
| 测试步骤 | * 初始化ROS节点 * 设置节点状态为Wait * 调用save\_map\_serve方法 |
| 期望输出 | * 返回SaveMapResponse(success=True) * 节点状态self.node\_info.state切换至Save\_Map并最终恢复到Wait * 发布SaveMap消息 |
| 测试成功标准 | 成功发布SaveMap消息并返回SaveMapResponse(success=True) |
| 用例备注 | 无 |

**（20）白盒测试用例20**

**表71 controller包pending节点单元测试白盒测试用例7说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-PEND-07 |
| 被测单元 | navigate\_serve |
| 优先级 | P0 |
| 测试输入 | NavigateGoal |
| 测试步骤 | * 初始化ROS节点 * 设置节点状态为Wait * 调用navigate\_serve方法 |
| 期望输出 | * 返回NavigateResult(result="success")或NavigateResult(result="fail") * 发布导航反馈消息NavigateFeedback |
| 测试成功标准 | 成功执行导航任务，发布相应的导航反馈并返回结果。 |
| 用例备注 | 无 |

**（21）白盒测试用例21**

**表71 controller包core节点单元测试白盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-01 |
| 被测单元 | handle\_start |
| 优先级 | P0 |
| 测试输入 | StartRequest(mode=1) |
| 测试步骤 | * 初始化ROS节点 * 调用handle\_start方法，并传入StartRequest(mode=1) |
| 期望输出 | * 返回StartResponse(success=True)。 * 调用/ctrl/pending/start服务。 |
| 测试成功标准 | 成功调用相应服务并返回StartResponse(success=True)。 |
| 用例备注 | 无 |

**（22）白盒测试用例22**

**表71 controller包core节点单元测试白盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-02 |
| 被测单元 | handle\_stop |
| 优先级 | P0 |
| 测试输入 | StopRequest(mode=1) |
| 测试步骤 | * 初始化ROS节点。 * 调用handle\_stop方法，并传入StopRequest(mode=1)。 |
| 期望输出 | * 返回StopResponse(success=True)。 * 调用/ctrl/pending/stop服务。 |
| 测试成功标准 | 成功调用相应服务并返回StopResponse(success=True)。 |
| 用例备注 | 无 |

**（23）白盒测试用例23**

**表71 controller包core节点单元测试白盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-03 |
| 被测单元 | handle\_change\_mode |
| 优先级 | P0 |
| 测试输入 | ChangeModeRequest(mode=2) |
| 测试步骤 | * 初始化ROS节点。 * 设置当前模式为1。 * 调用handle\_change\_mode方法，并传入ChangeModeRequest(mode=2)。 |
| 期望输出 | * 返回ChangeModeResponse(success=True)。 * 调用/ctrl/pending/stop服务，切换模式为2。 * 发布Info消息。 |
| 测试成功标准 | 成功切换模式并返回ChangeModeResponse(success=True)。 |
| 用例备注 | 无 |

**（24）白盒测试用例24**

**表71 controller包core节点单元测试白盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-04 |
| 被测单元 | scram\_callback |
| 优先级 | P1 |
| 测试输入 | ScramRequest(active=True) |
| 测试步骤 | * 初始化ROS节点。 * 调用scram\_callback方法，并传入ScramRequest(active=True)。 |
| 期望输出 | * 返回ScramResponse(success=True)。 * 发布Info消息，scram状态为True。 |
| 测试成功标准 | 成功更新急停状态并返回ScramResponse(success=True)。 |
| 用例备注 | 无 |

**（25）白盒测试用例25**

**表71 controller包core节点单元测试白盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-05 |
| 被测单元 | \_cmd\_vel\_callback |
| 优先级 | P1 |
| 测试输入 | Twist |
| 测试步骤 | * 初始化ROS节点。 * 设置self.scram为True。 * 调用\_cmd\_vel\_callback方法，并传入Twist消息。 |
| 期望输出 | 发布Twist消息，线速度和角速度均为0。 |
| 测试成功标准 | 成功发布急停Twist消息。 |
| 用例备注 | 无 |

**（26）白盒测试用例26**

**表71 controller包core节点单元测试白盒测试用例6说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-06 |
| 被测单元 | \_cmd\_vel\_callback |
| 优先级 | P1 |
| 测试输入 | Twist |
| 测试步骤 | * 初始化ROS节点。 * 设置self.scram为False。 * 调用\_cmd\_vel\_callback方法，并传入Twist消息。 |
| 期望输出 | 发布输入的Twist消息。 |
| 测试成功标准 | 成功发布输入的Twist消息。 |
| 用例备注 | 无 |

**（27）白盒测试用例27**

**表71 controller包core节点单元测试白盒测试用例7说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-07 |
| 被测单元 | handle\_start - 错误模式 |
| 优先级 | P0 |
| 测试输入 | StartRequest(mode=2) (当前模式为1) |
| 测试步骤 | * 初始化ROS节点。 * 设置当前模式为1。 * 调用handle\_start方法，并传入StartRequest(mode=2)。 |
| 期望输出 | 返回StartResponse(success=False)。 |
| 测试成功标准 | 当前模式与请求模式不一致时，返回StartResponse(success=False)。 |
| 用例备注 | 无 |

**（28）白盒测试用例28**

**表71 controller包core节点单元测试白盒测试用例8说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-08 |
| 被测单元 | handle\_stop - 错误模式 |
| 优先级 | P0 |
| 测试输入 | StopRequest(mode=2) (当前模式为1) |
| 测试步骤 | * 初始化ROS节点。 * 设置当前模式为1。 * 调用handle\_stop方法，并传入StopRequest(mode=2)。 |
| 期望输出 | 返回StopResponse(success=False)。 |
| 测试成功标准 | 当前模式与请求模式不一致时，返回StopResponse(success=False)。 |
| 用例备注 | 无 |

**（29）白盒测试用例29**

**表71 controller包core节点单元测试白盒测试用例9说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-09 |
| 被测单元 | handle\_change\_mode - 错误模式 |
| 优先级 | P0 |
| 测试输入 | ChangeModeRequest(mode=7) (无效模式) |
| 测试步骤 | * 初始化ROS节点。 * 设置当前模式为1。 * 调用handle\_change\_mode方法，并传入ChangeModeRequest(mode=7)。 |
| 期望输出 | 返回ChangeModeResponse(success=False)。 |
| 测试成功标准 | 请求模式无效时，返回ChangeModeResponse(success=False)。 |
| 用例备注 | 无 |

**（30）白盒测试用例30**

**表71 controller包core节点单元测试白盒测试用例10说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-CT-CORE-10 |
| 被测单元 | scram\_callback - 解除急停 |
| 优先级 | P1 |
| 测试输入 | ScramRequest(active=False) |
| 测试步骤 | * 初始化ROS节点。 * 调用scram\_callback方法，并传入ScramRequest(active=False)。 |
| 期望输出 | * 返回ScramResponse(success=True)。 * 发布Info消息，scram状态为False。 |
| 测试成功标准 | 成功解除急停状态并返回ScramResponse(success=True)。 |
| 用例备注 | 无 |

#### navigation Package

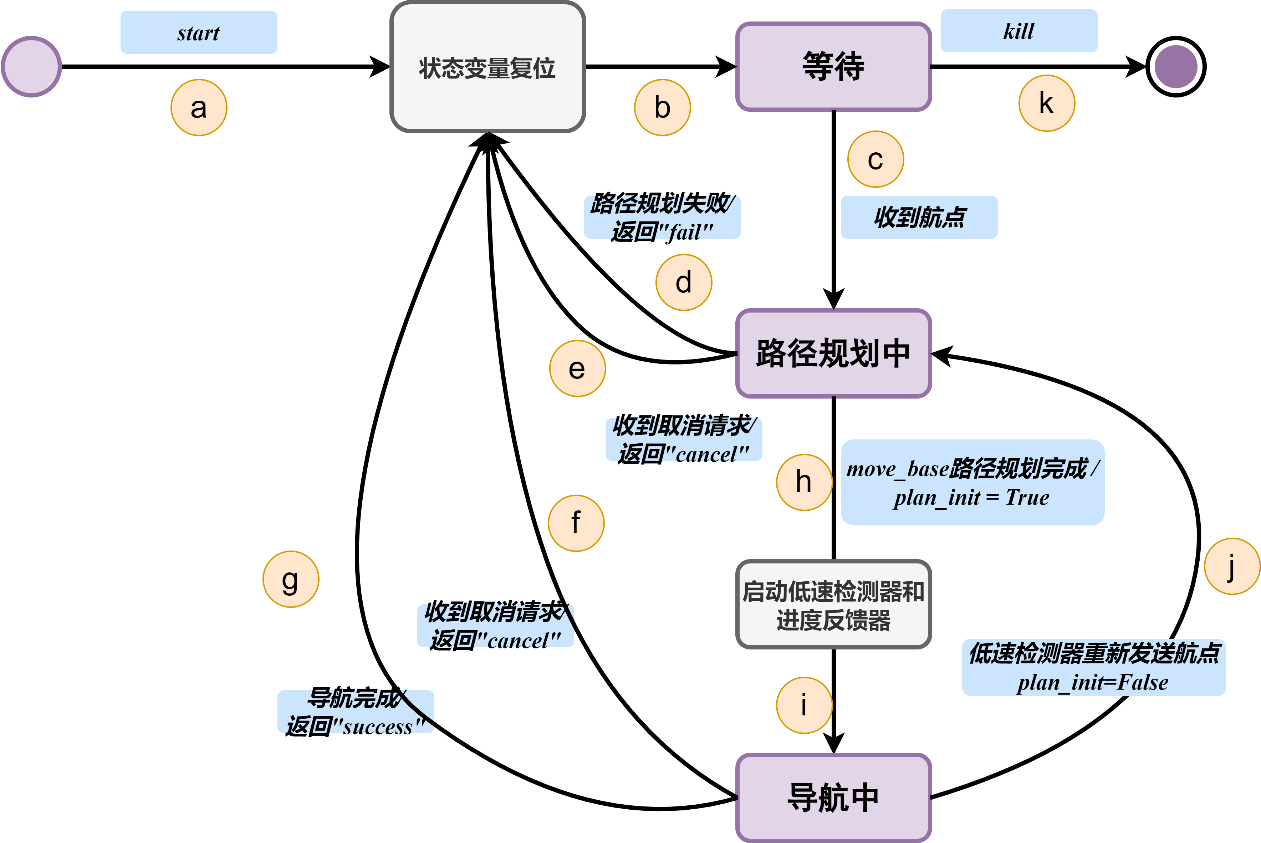


图 4.5 navigation状态转移图

**表79 navigation包单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROS-NAV |
| 相关需求 | 导航需求 |
| 测试涉及对象 | navigation功能包 |
| 前置条件 | 条件1：机器人电源开启，并且机器人状态无异常  条件2：机器人雷达启动，并且能够正常返回雷达识别到的数据  条件3：simple\_goal节点正常启动 |

**（1）白盒测试用例1**

**表80 navigation包单元测试白盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-NAV-01 |
| 优先级 | P2 |
| 测试输入 | * NavigateGoal对象，航点可达 |
| 测试步骤 | * 模拟客户端节点发送航点目标NavigateGoal（包含位置和最终朝向） * 启动simple\_goal节点和move\_base节点 * 观察模拟客户端收到的导航反馈和导航结果 |
| 期望输出 | 模拟客户端不断收到与实际导航进度匹配的导航进度反馈(feedback.percentage),并且处于正常状态(feedback.cur\_state="normal")；导航结束后收到导航成功的结果("success")。 |
| 测试成功标准 | 同期望输出。 |
| 用例备注 | 导航NavigateAction发送目标正常，过程反馈和结果反馈正常 |

**（2）白盒测试用例2**

**表81 navigation包单元测试白盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-NAV-02 |
| 优先级 | P2 |
| 测试输入 | * NavigateGoal对象，且该目标航点不可达 |
| 测试步骤 | * 模拟客户端节点发送一个不可达的航点目标NavigateGoal（包含位置和最终朝向） * 启动simple\_goal节点和move\_base节点 * 观察模拟客户端收到的导航反馈和导航结果 |
| 期望输出 | 模拟客户端收到导航过程反馈一直为“路径规划中”（"planning"），一段时间后收到导航失败("fail")的结果 |
| 测试成功标准 | 同期望输出。 |
| 用例备注 | 导航NavigateAction对于异常情况的过程反馈和结果返回正常。 |

**（3）白盒测试用例3**

**表82 navigation包单元测试白盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-NAV-03 |
| 优先级 | P2 |
| 测试输入 | * NavigateGoal对象，NavigateCancel消息 |
| 测试步骤 | * 模拟客户端节点发送一个航点目标NavigateGoal（包含位置和最终朝向） * 启动simple\_goal节点和move\_base节点 * 在到达航点之前，模拟客户端发送NavigateCancel消息 * 观察模拟客户端收到的导航反馈和导航结果 |
| 期望输出 | 在发送cancel后模拟客户端收到的导航结果应为“取消”（"cancel"） |
| 测试成功标准 | 同期望输出。 |
| 用例备注 | 导航NavigateAction的取消功能实现正常。 |

#### map\_provider Package

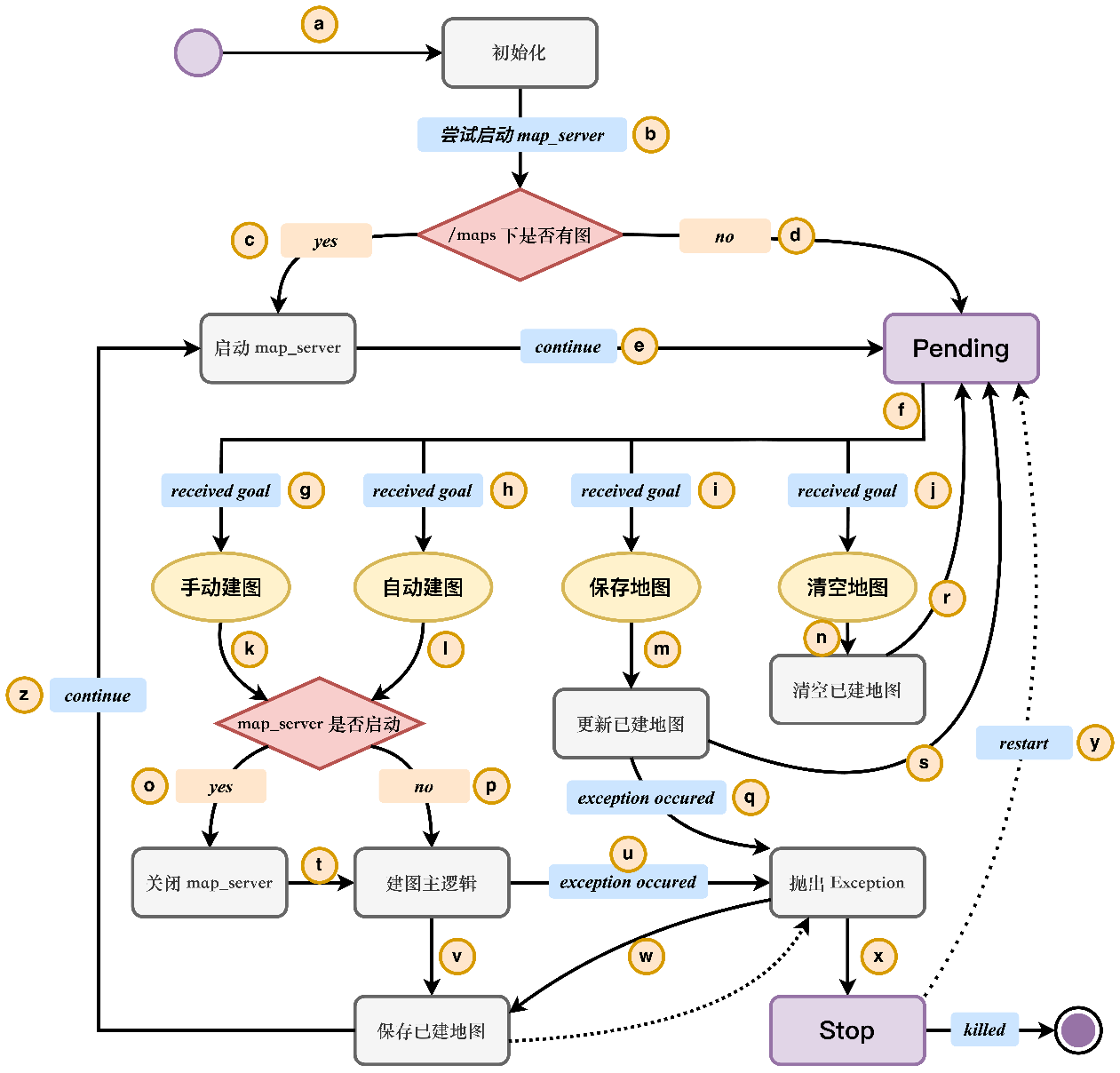
****

图 4.6 map\_provider白盒测试状态转移图

**表83 map\_provider包单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROS-MP |
| 相关需求 | 地图管理与创建需求 |
| 测试涉及对象 | map\_provider功能包 |
| 前置条件 | 条件1：机器人电源开启，并且机器人状态无异常  条件2：机器人雷达启动，并且能够正常返回雷达识别到的数据  条件3：map\_provider\_node正常启动 |

**（1）白盒测试用例1**

**表84 map\_provider包单元测试白盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-MP-01 |
| 优先级 | P2 |
| 测试输入 | * 空地图初始化数据 * 手动建图manual\_init\_map的action.goal |
| 测试步骤 | * 发送manual\_init\_map的action.goal * 启动 /slam\_gmapping * 在rviz中查看是否有周围地图信息 * 控制机器人运动，查看是否记录地图信息 |
| 期望输出 | 成功完成建图，并存储地图信息 |
| 测试成功标准 | 完成建图并存储地图信息，且能够在rviz或前端成功显示 |
| 用例备注 | 此测试用例验证了手动建图manual\_init\_map是否正常工作 |

**（2）白盒测试用例2**

**表85 map\_provider包单元测试白盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-MP-02 |
| 优先级 | P2 |
| 测试输入 | * 空地图初始化数据 * 自动建图auto\_init\_map的action.goal |
| 测试步骤 | * 发送auto\_init\_map的action.goal * 启动基于RRT的自动建图算法 * 在rviz中查看是否有周围地图信息 * 控制机器人运动，查看是否记录地图信息 |
| 期望输出 | 成功完成建图，并存储地图信息 |
| 测试成功标准 | 完成建图并存储地图信息，且能够在rviz或前端成功显示 |
| 用例备注 | 此测试用例验证了自动建图auto\_init\_map是否正常工作 |

**（3）白盒测试用例3**

**表86 map\_provider包单元测试白盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-MP-03 |
| 优先级 | P2 |
| 测试输入 | * 已经构建完成的地图数据 |
| 测试步骤 | * 发送SaveMap.msg，调用save\_map尝试保存地图 * 检查/map\_provider/maps下是否有.pgm和.yaml的地图文件 |
| 期望输出 | /map\_provider/maps下存在.pgm和.yaml的地图文件，且能够正确加载 |
| 测试成功标准 | 地图文件成功保存、格式正确且能正确重新加载 |
| 用例备注 | 此测试用例验证了保存地图save\_map是否正常工作 |

**（4）白盒测试用例4**

**表87 map\_provider包单元测试白盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-MP-04 |
| 优先级 | P2 |
| 测试输入 | * 空地图初始化数据 * 自动建图，auto\_init\_map的action.goal |
| 测试步骤 | * 启动自动建图 * 等待机器人启动建图后，设置路径障碍，使得无全局可达路径 * 观察并记录系统行为 |
| 期望输出 | * 机器人中止建图并报告异常，且成功存储已有地图 |
| 测试成功标准 | * 机器人成功中止建图并报告异常 |
| 用例备注 | 自动建图的异常处理 |

#### object\_detect Package

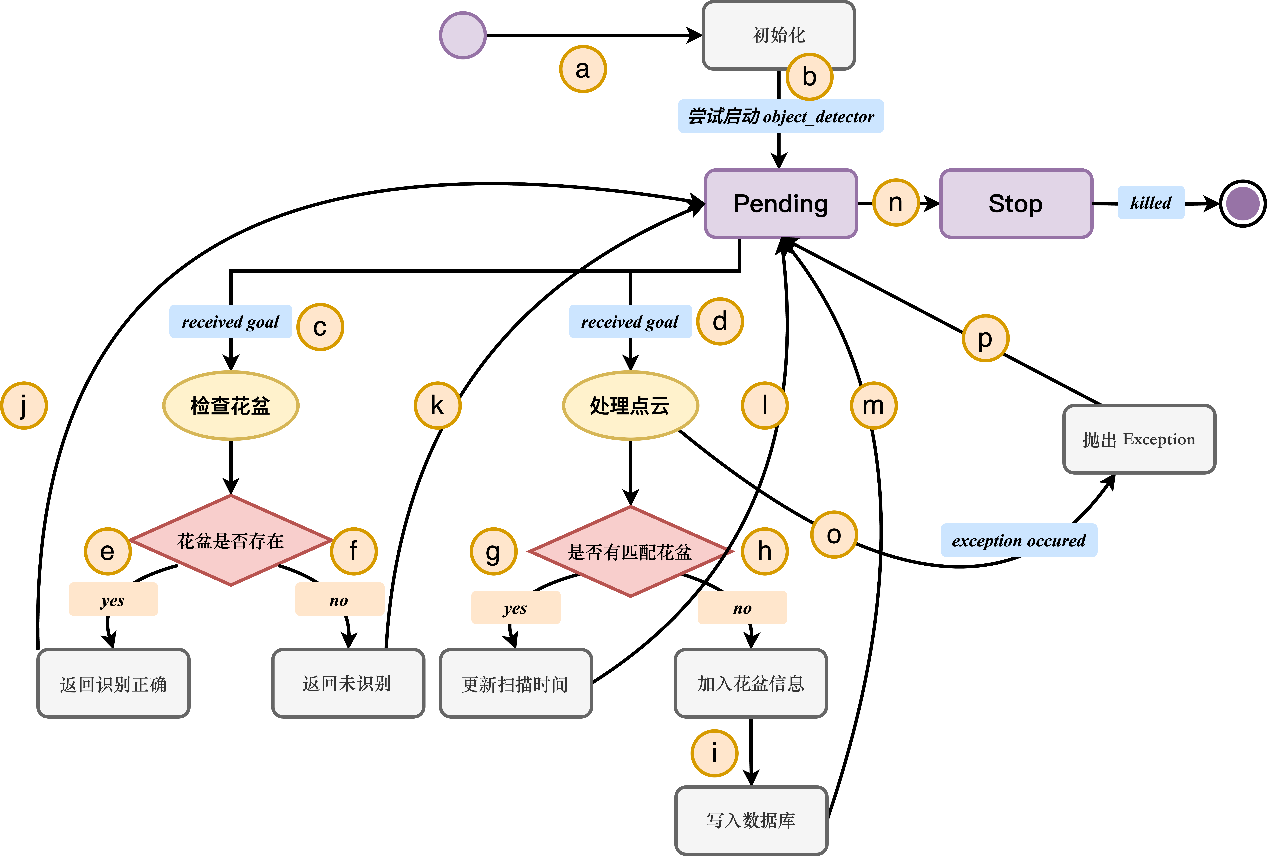


图 4.7 object\_detect白盒测试状态图

**表88 object\_detect包单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROS-OD |
| 相关需求 | 花盆识别需求 |
| 测试涉及对象 | 花盆识别节点 |
| 前置条件 | 条件1：机器人电源开启，并且机器人状态无异常  条件2：机器人摄像头启动，并且能够正常返回摄像头识别到的数据  条件3：花盆识别节点正常启动 |

**（1）白盒测试用例1**

**表89 object\_detect包单元测试白盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-OD-01 |
| 被测单元 | handle\_check\_pot |
| 优先级 | P0 |
| 测试输入 | CheckPotRequest(id=1) |
| 测试步骤 | * 初始化ROS节点。 * 设置花盆ID为1，last\_scan\_time为当前时间。 * 调用handle\_check\_pot方法，并传入CheckPotRequest(id=1)。 |
| 期望输出 | 返回CheckPotResponse(success=True)。 |
| 测试成功标准 | 当花盆存在且未超时时，返回CheckPotResponse(success=True)。 |
| 用例备注 |  |

**（2）白盒测试用例2**

**表89 object\_detect包单元测试白盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-OD-02 |
| 被测单元 | handle\_check\_pot |
| 优先级 | P0 |
| 测试输入 | CheckPotRequest(id=2) |
| 测试步骤 | * 初始化ROS节点。 * 设置pots字典，包含一个花盆ID为2，其最后扫描时间为当前时间减去6秒。 * 调用handle\_check\_pot方法，并传入CheckPotRequest(id=2)。 |
| 期望输出 | 返回CheckPotResponse(success=False)。 |
| 测试成功标准 | 当花盆存在但已超时时，返回CheckPotResponse(success=False)。 |
| 用例备注 | 无 |

**（3）白盒测试用例3**

**表89 object\_detect包单元测试白盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-OD-03 |
| 被测单元 | cmd\_vel\_callback |
| 优先级 | P1 |
| 测试输入 | Twist消息，linear和angular速度均为0.1。 |
| 测试步骤 | 初始化ROS节点。  调用cmd\_vel\_callback方法，并传入Twist消息，linear和angular速度均为0.1。 |
| 期望输出 | current\_linear\_velocity和current\_angular\_velocity均为0.1。 |
| 测试成功标准 | 成功更新当前线速度和角速度。 |
| 用例备注 |  |

**（4）白盒测试用例4**

**表89 object\_detect包单元测试白盒测试用例说4明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-OD-04 |
| 被测单元 | image\_callback |
| 优先级 | P1 |
| 测试输入 | Image消息 |
| 测试步骤 | * 初始化ROS节点。 * 调用image\_callback方法，并传入Image消息。 |
| 期望输出 | latest\_image为输入的Image消息。 |
| 测试成功标准 | 成功缓存最新的图像数据。 |
| 用例备注 | 无 |

**（5）白盒测试用例5**

**表89 object\_detect包单元测试白盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-OD-05 |
| 被测单元 | handle\_update\_pots |
| 优先级 | P0 |
| 测试输入 | PointStamped和PointCloud2消息，机器人速度为0 |
| 测试步骤 | * 初始化ROS节点。 * 设置当前线速度和角速度为0。 * 调用handle\_update\_pots方法，并传入PointStamped和PointCloud2消息。 |
| 期望输出 | 更新或新增花盆信息到pots字典中。 |
| 测试成功标准 | 成功更新或新增花盆信息。 |
| 用例备注 | 无 |

**（6）白盒测试用例6**

**表89 object\_detect包单元测试白盒测试用例6说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-OD-06 |
| 被测单元 | handle\_update\_pots |
| 优先级 | P0 |
| 测试输入 | PointStamped和PointCloud2消息，机器人速度不为0 |
| 测试步骤 | * 初始化ROS节点。 * 设置当前线速度和角速度为0.2。 * 调用handle\_update\_pots方法，并传入PointStamped和PointCloud2消息。 |
| 期望输出 | 不更新花盆信息到pots字典中。 |
| 测试成功标准 | 在机器人移动时，不更新花盆信息。 |
| 用例备注 | 无 |

**（7）白盒测试用例7**

**表89 object\_detect包单元测试白盒测试用例7说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-OD-07 |
| 被测单元 | handle\_update\_pots - 匹配现有花盆 |
| 优先级 | P0 |
| 测试输入 | PointStamped和PointCloud2消息，机器人速度为0，匹配现有花盆 |
| 测试步骤 | * 初始化ROS节点。 * 设置当前线速度和角速度为0。 * 设置pots字典，包含一个与PointStamped位置相近的花盆。 * 调用handle\_update\_pots方法，并传入PointStamped和PointCloud2消息。 |
| 期望输出 | 更新现有花盆的last\_scan\_time。 |
| 测试成功标准 | 成功更新现有花盆的last\_scan\_time。 |
| 用例备注 | 无 |

**（8）白盒测试用例8**

**表89 object\_detect包单元测试白盒测试用例8说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-OD-08 |
| 被测单元 | handle\_update\_pots - 新增花盆 |
| 优先级 | P0 |
| 测试输入 | PointStamped和PointCloud2消息，机器人速度为0，不匹配现有花盆 |
| 测试步骤 | * 初始化ROS节点。 * 设置当前线速度和角速度为0。 * 确保pots字典为空或不包含与PointStamped位置相近的花盆。 * 调用handle\_update\_pots方法，并传入PointStamped和PointCloud2消息。 |
| 期望输出 | 新增花盆信息到pots字典中。 |
| 测试成功标准 | 成功新增花盆信息。 |
| 用例备注 | 无 |

**（9）白盒测试用例9**

**表89 object\_detect包单元测试白盒测试用例9说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-OD-09 |
| 被测单元 | handle\_update\_pots - 无法转换坐标 |
| 优先级 | P0 |
| 测试输入 | PointStamped和PointCloud2消息，机器人速度为0，无法转换坐标 |
| 测试步骤 | * 初始化ROS节点。 * 设置当前线速度和角速度为0。 * 模拟tf转换失败。 * 调用handle\_update\_pots方法，并传入PointStamped和PointCloud2消息。 |
| 期望输出 | 不更新或新增花盆信息到pots字典中。 |
| 测试成功标准 | 坐标转换失败时，不更新或新增花盆信息。 |
| 用例备注 | 无 |

#### robot\_arm Package

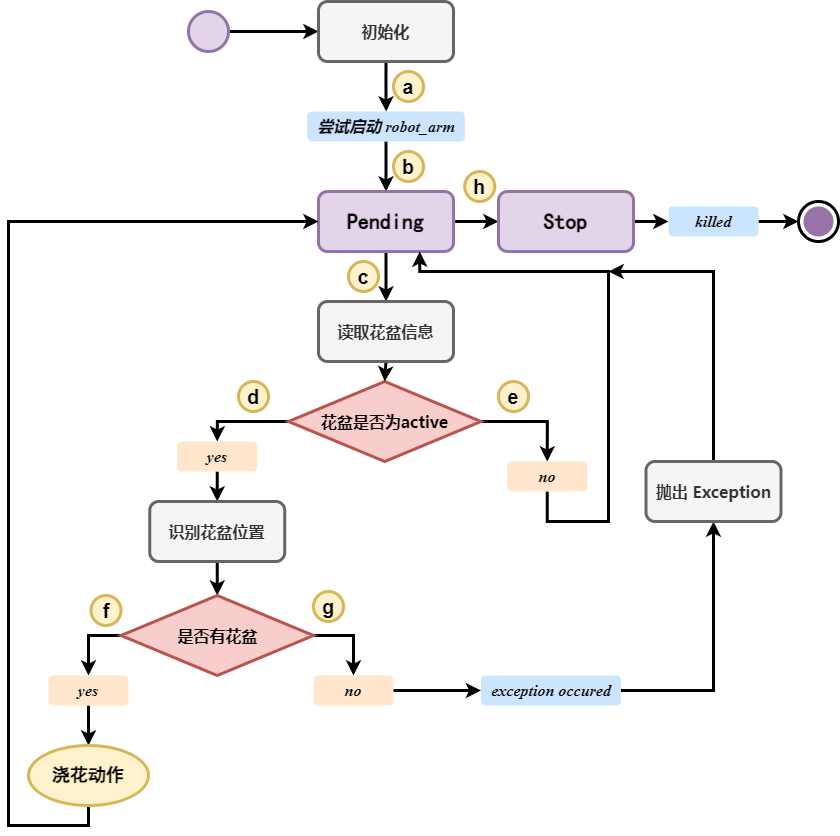


图 4.8 robot\_arm白盒测试状态转移图

**表92 robot\_arm包单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROS-RA |
| 相关需求 | 浇花需求 |
| 测试涉及对象 | 机械臂控制节点 |
| 前置条件 | 条件1：机器人电源开启，并且机器人状态无异常  条件2：机器人机械臂正常启动，且机械臂移动范围内无阻碍  条件3：机械臂控制节点正常启动 |

**（1）白盒测试用例1**

**表89 robot\_arm包单元测试白盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-RA-01 |
| 被测单元 | executeCallback |
| 优先级 | P2 |
| 测试输入 | 花盆三维坐标 (1.0, 2.0, 1.2) |
| 测试步骤 | * 将花盆摆放在机器人视野范围内 * 发送AimGoal引导机器人自身矫正位置，开始执行浇水动作 * 观察浇水范围是否精确，是否存在明显的水分浪费现象 |
| 期望输出 | 机器人控制喷水软管对准花盆，并为水泵通电开始浇水 |
| 测试成功标准 | 浇水范围覆盖目标花盆范围，且无明显偏差，无明显水分浪费现象 |
| 用例备注 | 该测试用例验证了机器人可正常识别花盆并执行浇花动作 |

**（2）白盒测试用例2**

**表89 robot\_arm包单元测试白盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-RA-02 |
| 被测单元 | executeCallback |
| 优先级 | P2 |
| 测试输入 | * 花盆三维坐标 (1.0, 2.0, 1.2) * 水源不足的水泵 |
| 测试步骤 | * 将花盆摆放在机器人视野范围内，同时清空水泵的水源 * 发送AimGoal引导机器人自身矫正位置，准备执行浇水动作 * 观察并记录机器人系统行为 |
| 期望输出 | 机器人停止执行浇花动作，并上报“水泵缺水异常” |
| 测试成功标准 | 机器人停止执行浇水动作，并成功上报异常 |
| 用例备注 | 该测试用例验证了机器人可在浇花前正常处理“水泵缺水异常” |

**（3）白盒测试用例3**

**表89 robot\_arm包单元测试白盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-RA-03 |
| 被测单元 | executeCallback |
| 优先级 | P1 |
| 测试输入 | 无花盆 |
| 测试步骤 | * 临时撤走原先花盆，模拟花盆消失 * 发送AimGoal引导机器人矫正自身位置，准备执行浇水动作 * 观察并记录机器人系统行为 |
| 期望输出 | 机器人停止浇花动作，并上报“花盆缺失异常” |
| 测试成功标准 | 机器人停止浇花并成功上报异常 |
| 用例备注 | 该测试用例验证了机器人在浇花前可正确处理“花盆缺失”异常 |

**（4）白盒测试用例4**

**表89 robot\_arm包单元测试白盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-RA-04 |
| 被测单元 | executeCallback |
| 优先级 | P1 |
| 测试输入 | * AimGoal(id=1) * 数据库中存在ID为1且为active的花盆 |
| 测试步骤 | * 模拟数据库中存在ID为1且为active的花盆 * 发送AimGoal(id=1)以开始执行浇水动作 * 观察机器人的动作执行过程和日志输出 |
| 期望输出 | 机器人执行矫正、伸展机械臂、浇水和收回机械臂的动作 |
| 测试成功标准 | 机器人成功执行所有动作，并返回AimResult(success=True, info="the pot is active!") |
| 用例备注 | 该测试用例验证了机器人在花盆处于active状态下的浇水流程 |

**（5）白盒测试用例5**

**表89 robot\_arm包单元测试白盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-RA-05 |
| 被测单元 | executeCallback |
| 优先级 | P1 |
| 测试输入 | * AimGoal(id=2) * 数据库中存在ID为2但为inactive的花盆 |
| 测试步骤 | * 模拟数据库中存在ID为2但为inactive的花盆 * 发送AimGoal(id=2)以开始执行浇水动作 * 观察机器人的动作执行过程和日志输出 |
| 期望输出 | 机器人不执行浇水动作，并返回AimResult(success=True, info="the pot is not active...") |
| 测试成功标准 | 机器人未执行浇水动作，并返回正确的结果信息 |
| 用例备注 | 无 |

**（6）白盒测试用例6**

**表89 robot\_arm包单元测试白盒测试用例6说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-RA-06 |
| 被测单元 | executeCallback |
| 优先级 | P1 |
| 测试输入 | * AimGoal(id=3) * 数据库服务不可用 |
| 测试步骤 | * 模拟数据库服务不可用的情况 * 发送AimGoal(id=3)以开始执行浇水动作 * 观察机器人的日志输出 |
| 期望输出 | 机器人日志输出服务调用失败信息，并返回AimResult(success=False, info="fail to get pot infomation...") |
| 测试成功标准 | 机器人未执行浇水动作，并返回正确的错误信息 |
| 用例备注 | 该测试用例验证了机器人在数据库服务不可用情况下的行为 |

#### pot\_database Package

**表96 pot\_database包单元测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-ROS-DB |
| 相关需求 | 花盆信息存取需求 |
| 测试涉及对象 | 数据库节点 |
| 前置条件 | 条件1：机器人电源开启，并且机器人状态无异常  条件2：花盆节点正常启动 |

**（1）白盒测试用例1**

**表97 pot\_database包单元测试白盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-DB-01 |
| 优先级 | P1 |
| 测试输入 | 若干个具有特定 ID 和详细信息的 PotInfo 对象 |
| 测试步骤 | * 调用单元测试中 test\_list\_pots 方法 * 在该方法中，创建并添加若干个花盆信息 * 调用列表服务获取花盆列表 |
| 期望输出 | 获取的花盆列表中包含所有花盆的信息 |
| 测试成功标准 | test\_list\_pots 方法中的断言全部通过，确保所有花盆的信息均已经返回 |
| 用例备注 | 测试获取花盆列表功能 |

**（2）白盒测试用例2**

**表98 pot\_database包单元测试白盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-DB-02 |
| 优先级 | P1 |
| 测试输入 | 一个具有特定 ID 和详细信息的 PotInfo 对象 |
| 测试步骤 | * 调用 test\_set\_and\_get\_pot\_info 方法 * 在方法中设置一个花盆的信息，并调用添加花盆的服务 * 获取该花盆信息并验证详细信息 |
| 期望输出 | 能够正确地设置并返回指定的花盆信息 |
| 测试成功标准 | test\_set\_and\_get\_pot\_info方法中的所有断言均通过，设置和获取的花盆信息匹配 |
| 用例备注 | 测试获取花盆列表功能，确保服务能正确返回存储的花盆信息 |

**（3）白盒测试用例3**

**表99 pot\_database包单元测试白盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-DB-03 |
| 优先级 | P1 |
| 测试输入 | 激活指令以及目标花盆的ID |
| 测试步骤 | * 调用 test\_set\_active 方法 * 发送激活请求给指定ID的花盆 |
| 期望输出 | 花盆的激活状态被成功修改 |
| 测试成功标准 | test\_set\_active中所有断言均通过，返回的激活状态修改成功 |
| 用例备注 | 测试花盆激活功能，确保激活状态可以被正确修改 |

**（4）白盒测试用例4**

**表100 pot\_database包单元测试白盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-ROS-DB-04 |
| 优先级 | P1 |
| 测试输入 | 待删除的花盆ID |
| 测试步骤 | * 调用 test\_delete\_pot 方法 * 创建并设置两个花盆信息，将两个花盆的信息添加到数据库中 * 删除其中一个花盆 * 获取并验证当前花盆列表 |
| 期望输出 | 指定ID的花盆被成功删除，花盆列表更新 |
| 测试成功标准 | test\_delete\_pot方法中所有断言均通过，列表中不再包含被删除的花盆，且仍正确列出其他花盆 |
| 用例备注 | 测试删除功能，确保可以正确地从系统中移除指定花盆信息 |

### 集成测试：功能需求

#### 4.3.4.1 建图

**表101 建图集成测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-INT-MapInit |
| 相关需求 | 建图需求（自动建图、手动建图） |
| 测试涉及对象 | 前端，core，auto\_map，manual\_map, map\_provider，object\_detect |

**（1）黑盒测试用例1**

**表102 建图集成测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-MapInit-01 |
| 优先级 | P0 |
| 被测功能 | 手动建图 |
| 测试输入 | * 简单真实场景 * 启动手动建图功能 |
| 测试步骤 | * 在前端触发手动建图功能，同时用手柄或键盘控制机器人运行 |
| 期望输出/行为 | * 前端可以看到已建立的地图信息 |
| 测试成功标准/评价 | * 机器人成功建图，且地图成功存储 |
| 用例备注 | 简单场景下手动建图功能测试。  机器人处于等待(pending)状态。 |

**（2）黑盒测试用例2**

**表103建图集成测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-MapInit-02 |
| 优先级 | P0 |
| 被测功能 | | 自动建图功能 |
| 测试输入 | * 简单真实场景 * 启动自动建图功能 |
| 测试步骤 | * 在前端触发自动建图功能 |
| 期望输出/行为 | * 机器人开始自动运行 * 前端可以看到已建立的地图信息 |
| 测试成功标准/评价 | * 机器人成功建图，且地图成功存储 |
| 用例备注 | 简单场景下自动建图功能测试。  机器人处于等待(pending)状态。 |

**（3）黑盒测试用例3**

**表104 建图集成测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-MapInit-03 |
| 被测功能 | 自动建图功能 |
| 优先级 | P0 |
| 测试输入 | * 复杂环境场景 * 启动自动建图功能 |
| 测试步骤 | * 在前端触发自动建图功能 |
| 期望输出/行为 | * 机器人开始自动运行 * 前端可以看到已建立的地图信息 |
| 测试成功标准/评价 | * 机器人成功建图，且地图成功存储 |
| 用例备注 | 复杂场景下自动建图功能测试。  机器人处于等待(pending)状态。 |

**（4）黑盒测试用例4**

**表105建图集成测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-MapInit-04 |
| 优先级 | P0 |
| 被测功能 | 自动建图功能 |
| 测试输入 | * 无可达路径场景 * 启动自动建图功能 |
| 测试步骤 | * 在前端触发自动建图功能 |
| 期望输出/行为 | * 机器人开始自动运行，随后中止并抛出异常 * 前端可以看到已建立的局部地图信息，但无全局地图信息 |
| 测试成功标准/评价 | * 机器人成功启动并报告异常 |
| 用例备注 | 无可达路径场景下自动建图功能测试。  机器人处于等待(pending)状态。 |

#### 4.3.4.2 导航与动态避障

**表106 导航与动态避障集成测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-INT-Navigation |
| 相关需求 | 导航与动态避障需求 |
| 测试涉及对象 | 前端，core，navigation，object\_detect |

**（1）黑盒测试用例1**

**表107 导航与动态避障集成测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-Navigation-01 |
| 优先级 | P0 |
| 被测功能 | 导航与动态避障功能 |
| 测试输入 | * 简单无障碍地图 * 前端发送的航点信息 |
| 测试步骤 | * 前端选定航点，开始导航 |
| 期望输出/行为 | * 机器人成功抵达航点 * 前端获得导航成功的反馈 |
| 测试成功标准/评价 | * 机器人成功抵达航点 |
| 用例备注 | 简单无障碍场景下的导航测试。  机器人处于等待(pending)状态。 |

**（2）黑盒测试用例2**

**表108 导航与动态避障集成测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-Navigation-02 |
| 优先级 | P0 |
| 被测功能 | 导航与动态避障功能 |
| 测试输入 | * 简单、单障碍地图 * 前端发送的航点信息 |
| 测试步骤 | * 前端选定航点，开始导航 |
| 期望输出/行为 | * 机器人成功抵达航点 * 前端获得导航成功的反馈 |
| 测试成功标准/评价 | 机器人成功抵达航点 |
| 用例备注 | 单障碍场景下的导航测试。  机器人处于等待(pending)状态。 |

**（3）黑盒测试用例3**

**表109 导航与动态避障集成测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-Navigation-03 |
| 优先级 | P0 |
| 被测功能 | 导航与动态避障功能 |
| 测试输入 | * 复杂、长距离地图 * 前端发送的航点信息 |
| 测试步骤 | * 前端选定航点，开始导航 |
| 期望输出/行为 | * 机器人成功抵达航点 * 前端获得导航成功的反馈 |
| 测试成功标准/评价 | * 机器人成功抵达航点 |
| 用例备注 | 复杂、长距离场景下的导航测试。  机器人处于等待(pending)状态。 |

**（4）黑盒测试用例4**

**表110 导航与动态避障集成测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-Navigation-04 |
| 优先级 | P0 |
| 被测功能 | 导航与动态避障功能 |
| 测试输入 | * 复杂、长距离地图 * 前端发送的航点信息 |
| 测试步骤 | * 前端选定航点，开始导航 |
| 期望输出/行为 | * 机器人成功抵达航点 * 前端获得导航成功的反馈 |
| 测试成功标准/评价 | * 机器人成功抵达航点 |
| 用例备注 | 复杂、长距离场景下的导航测试。  机器人处于等待(pending)状态。 |

#### 4.3.4.3 浇水

**表111 浇水集成测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-INT-WaterJet |
| 相关需求 | 浇水（定点浇水、自动巡航浇水） |
| 测试涉及对象 | 前端，core，auto\_water, navigation, object\_detect, robor\_arm |

**（1）黑盒测试用例1**

**表112浇水集成测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-WaterJet-01 |
| 优先级 | P0 |
| 被测功能 | 定点、单盆浇水功能 |
| 测试输入 | * 单花盆场景 * 前端指定的待浇水花盆 |
| 测试步骤 | * 前端选定浇水花盆 |
| 期望输出/行为 | * 前端得到反馈“成功浇水” * 机器人成功为选定花盆浇水 |
| 测试成功标准/评价 | * 成功为花盆浇水 |
| 用例备注 | 定点、单盆浇水功能测试。  机器人处于等待(pending)状态。 |

**（2）黑盒测试用例2**

**表113浇水集成测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-WaterJet-02 |
| 优先级 | P0 |
| 被测功能 | 多花盆浇水功能 |
| 测试输入 | * 多花盆场景 * 前端指定的待浇水花盆序列 |
| 测试步骤 | * 前端选定浇水花盆序列 |
| 期望输出/行为 | * 前端得到反馈“成功浇水” * 机器人成功为所有选定花盆浇水 |
| 测试成功标准/评价 | * 成功为所选花盆浇水 * 未选择花盆不会被浇水 |
| 用例备注 | 多花盆场景浇水功能测试。  机器人处于等待(pending)状态。 |

**（3）黑盒测试用例3**

**表114浇水集成测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-WaterJet-03 |
| 优先级 | P0 |
| 被测功能 | 自动巡航浇水功能 |
| 测试输入 | * 多花盆场景 * 前端设定“自动巡航浇水”模式 |
| 测试步骤 | * 前端启动自动巡航浇水 |
| 期望输出 | * 前端得到反馈“成功浇水” * 机器人自动巡检地图，为所需花盆浇水 |
| 测试成功标准 | * 成功为缺水花盆浇水 |
| 用例备注 | 自动巡航浇水功能测试。  机器人处于等待(pending)状态。 |

**（4）黑盒测试用例4**

**表115浇水集成测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-WaterJet-04 |
| 优先级 | P0 |
| 被测功能 | 无花盆场景下的异常处理功能 |
| 测试输入 | * 无花盆场景 * 前端指定的待浇水花盆 |
| 测试步骤 | * 前端选定浇水花盆 |
| 期望输出/行为 | * 前端得到反馈“未识别到目标花盆，暂停浇水” * 机器人中止浇水 |
| 测试成功标准/评价 | * 机器人成功中止浇水 |
| 用例备注 | 无花盆、异常功能测试。  机器人处于等待(pending)状态。 |

#### 4.3.4.4 启动服务

**表116 启动服务集成测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-INT-ServiceInit |
| 相关需求 | 全服务的启动 |
| 测试涉及对象 | 前端，core，all sub-modules |

**（1）黑盒测试用例1**

**表117 启动服务集成测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-ServiceInit-01 |
| 优先级 | P0 |
| 被测功能 | 启动服务功能 |
| 测试输入 | * 正常启动服务流程 |
| 测试步骤 | * 正常启动服务 |
| 期望输出/行为 | * 前端成功启动 * 后端成功启动 * 机器人等待响应 (pending) |
| 测试成功标准/评价 | * 所有服务成功启动 |
| 用例备注 | 正常服务启动流程。 |

**（2）黑盒测试用例2**

**表118 启动服务集成测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-ServiceInit-02 |
| 优先级 | P0 |
| 被测功能 | 启动服务异常处理功能 |
| 测试输入 | * 正常启动服务流程 * 设定缺失依赖 |
| 测试步骤 | * 正常启动服务 |
| 期望输出/行为 | * 系统中止启动 * 系统反馈所缺失的特定依赖 |
| 测试成功标准/评价 | * 系统成功中止启动，并报告异常 |
| 用例备注 | 异常服务启动流程。 |

**（3）黑盒测试用例3**

**表119 启动服务集成测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-INT-ServiceInit-03 |
| 优先级 | P0 |
| 被测功能 | 异常后重启服务功能 |
| 测试输入 | * 服务异常中断后，重启服务流程 |
| 测试步骤 | * 恢复依赖，允许系统重启 |
| 期望输出 | * 前端成功启动 * 后端成功启动 * 机器人等待响应 (pending) |
| 测试成功标准 | * 所有服务成功启动 |
| 用例备注 | 异常后重启服务流程。 |

### 系统测试：业务需求

#### 自动建图测试

**表120 自动建图测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-SYS-AutoMap |
| 相关需求 | 自动建图需求 |
| 测试涉及对象 | 前端，core，auto\_map，map\_provider，object\_detect |

**（1）黑盒测试用例1**

**表121 自动建图测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-AutoMap-01 |
| 优先级 | P0 |
| 测试类别 | 功能测试 |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击auto\_map按钮。 * 等待建图完成。   **监控指标：**   * 监控前端地图页面是否显示地图信息。 * 检查地图上是否能看到所有花盆的位置信息。   **异常处理：**   * 如果建图过程出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**  前端地图页面可以看到地图信息，并能看到所有花盆的位置信息 |
| 测试成功标准 | 自动建图基本功能正常。 |
| 用例备注 | 温室场景主要路径上无障碍。  机器人处于等待(pending)状态 |

**（2）黑盒测试用例2**

**表122 自动建图测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-AutoMap-02 |
| 优先级 | P0 |
| 测试类别 | 功能测试 |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保已有一个用自动建图保存的地图。 * 确保机器人处于等待(pending)状态。   **执行步骤：**  前端点击清除地图。  **监控指标：**  监控前端地图页面是否清除地图信息。**异常处理：**   * 如果清除地图过程出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**  前端地图信息清除 |
| 测试成功标准 | 自动建图的清除地图基本功能正常 |
| 用例备注 | 已经有一个用自动建图保存的地图  机器人处于等待(pending)状态 |

**（3）黑盒测试用例3**

**表123 自动建图单元测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-AutoMap-03 |
| 优先级 | P0 |
| 测试类别 | 功能测试（异常处理） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保温室场景主要通道上有障碍。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击auto\_map按钮。 * 等待反馈。   **监控指标：**   * 监控前端是否收到建图失败的反馈消息。 * 检查机器人是否退出自动建图状态。   **异常处理：**   * 如果未收到建图失败的反馈消息，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**  前端收到建图失败的反馈消息，并退出自动建图状态。 |
| 期望输出 | 前端收到建图失败的反馈消息，并退出自动建图状态。 |
| 测试成功标准 | 自动建图的异常处理功能正常 |
| 用例备注 | 温室场景主要通道上有障碍  机器人处于等待(pending)状态 |

**（4）黑盒测试用例4**

**表124 自动建图测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-AutoMap-04 |
| 优先级 | P0 |
| 测试类别 | 功能测试（实时取消） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击auto\_map按钮。 * 一段时间后点击取消自动建图按钮。   **监控指标：**   * 监控前端是否收到自动建图取消的反馈信息。 * 检查此次获得的部分地图是否没有覆盖原有地图。   **异常处理：**   * 如果未收到自动建图取消的反馈信息，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**  前端收到自动建图取消的反馈信息，并且此次获得的部分地图没有覆盖原有地图。 |
| 测试成功标准 | 自动建图的实时取消功能正常 |
| 用例备注 | 机器人处于等待(pending)状态 |

#### 手动建图测试

**表125 手动建图测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-SYS-ManualMap |
| 相关需求 | 手动建图需求 |
| 测试涉及对象 | 前端，core，manual\_map，map\_provider，object\_detect |

**（1）黑盒测试用例1**

**表126 手动建图测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-ManualMap-01 |
| 优先级 | P0 |
| 测试类别 | 功能测试 |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保温室中主要通道上无障碍。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击手动建图按钮。 * 通过手柄控制机器人移动，扫描整个温室。 * 前端点击保存地图按钮。   **监控指标：**   * 监控前端是否显示手动建图得到的地图信息。 * 检查地图上是否显示所有花盆的位置信息。   **异常处理：**   * 如果手动建图过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**  前端上显示手动建图得到的地图信息，包括所有花盆的位置信息。 |
| 测试成功标准 | 手动建图基本功能正常 |
| 用例备注 | 温室中主要通道上无障碍  机器人处于等待(pending)状态 |

**（2）黑盒测试用例2**

**表127 手动建图测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-ManualMap-02 |
| 优先级 | P0 |
| 测试类别 | 功能测试 |
| 测试流程 | * 前端点击清除地图按钮 |
| 测试步骤 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保已经保存了一个通过手动建图得到的地图。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击清除地图按钮。   **监控指标：**   * 监控前端上的地图信息是否被清除。   **异常处理：**   * 如果清除地图过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试成功标准 | 手动建图的清除地图功能正常 |
| 用例备注 | 已经保存了一个通过手动建图得到的地图 |

**（3）黑盒测试用例3**

**表128 手动建图测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-ManualMap-03 |
| 优先级 | P0 |
| 测试类别 | 功能测试（实时取消） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击手动建图按钮。 * 通过手柄控制机器人扫描一段时间。 * 前端点击取消手动建图按钮。   **监控指标：**   * 监控前端是否收到手动建图取消的反馈信息。 * 检查此次新得到的地图是否没有覆盖原有地图。   **异常处理：**   * 如果未收到手动建图取消的反馈信息，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**  前端收到手动建图取消的反馈信息，并且此次新得到的地图没有覆盖原有地图。 |
| 测试成功标准 | 手动建图的实时取消功能正常 |
| 用例备注 | 机器人处于等待(pending)状态 |

#### 巡检维护测试

**表129 巡检维护测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-SYS-Inspection |
| 相关需求 | 巡检维护需求 |
| 测试涉及对象 | 前端，core，inspection，pot\_database |

**（1）黑盒测试用例1**

**表130 巡检维护黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Inspection-01 |
| 优先级 | P0 |
| 测试类别 | 功能测试 |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保已经保存了一张地图，具有基本的花盆位置信息，但花盆含水量未初始化。   确保机器人处于等待(pending)状态。  **执行步骤：**   * 前端点击巡检维护按钮。 * 选择目标花盆并设置其含水量。   **监控指标：**   * 监控前端是否显示花盆含水量更新。 * 检查前端的花盆列表信息是否更新。   **异常处理：**   * 如果巡检维护过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**  花盆的含水量更新，前端的花盆列表信息更新。 |
| 测试成功标准 | 机器人设置花盆含水量功能正常 |
| 用例备注 | 已经保存了一张地图，具有基本的花盆位置信息，但花盆含水量未初始化。  机器人处于等待(pending)状态 |

#### 定点浇水测试

**表131 定点浇水测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-SYS-Target |
| 相关需求 | 定点浇水需求 |
| 测试涉及对象 | 前端，core，target，navigation，jet，object\_detect |

**（1）黑盒测试用例1**

**表132 定点浇水测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Target-01 |
| 优先级 | P0 |
| 测试类别 | 功能测试 |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中温室主要通道上无障碍，花盆没有消失，水泵含水量充足。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击定点浇水按钮。 * 前端选择待浇水的一组花盆。   **监控指标：**   * 监控机器人是否导航至目标花盆前方，并通过局部视觉校正将喷头对准花盆并浇水。 * 检查前端是否显示浇花进度反馈。 * 检查花盆含水量是否更新。   **异常处理：**   * 如果定点浇水过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**   * 机器人导航至目标花盆前方，并通过局部视觉校正将喷头对准花盆，并浇水。重复该过程至浇完最后一盆花。 * 在过程中能在前端看到浇花进度反馈。 * 花盆含水量更新。 |
| 测试成功标准 | 机器人定点浇水基本功能正常 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中温室主要通道上无障碍，花盆没有消失，水泵含水量充足。  机器人处于等待(pending)状态 |

**（2）黑盒测试用例2**

**表133 定点浇水测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Target-02 |
| 优先级 | P0 |
| 测试类别 | 功能测试（异常处理） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中温室主要通道上无障碍，但目标花盆A被移除，水泵含水量充足。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 在建图后搬走一盆花A。 * 前端点击定点浇水按钮。 * 前端选择待浇水的一组花盆，其中包含目标花盆A。   **监控指标：**   * 监控机器人导航至被移除的花盆A的位置时是否抛出花盆消失的异常。 * 检查前端是否显示该异常反馈。 * 监控机器人抛出异常后是否继续正常进行定点浇水。 * 检查除花盆A外的其余目标花盆含水量是否更新。   **异常处理：**   * 如果处理花盆消失异常过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**   * 当机器人导航至被移除的花盆A的位置时，抛出花盆消失的异常，前端能看到该异常反馈。 * 抛出异常后继续正常进行定点浇水。 * 除花盆A的含水量不变，其余目标花盆含水量更新。 |
| 测试成功标准 | 机器人定点浇水的处理花盆消失异常的功能正常。 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中温室主要通道上无障碍，但目标花盆A被移除，水泵含水量充足。  机器人处于等待(pending)状态 |

**（3）黑盒测试用例3**

**表134 定点浇水测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Target-03 |
| 优先级 | P0 |
| 测试类别 | 功能测试（实时取消） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中温室主要通道上无障碍，花盆没有消失，水泵含水量充足。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击定点浇水按钮。 * 前端选择待浇水的一组花盆。 * 在执行的一部分定点浇水任务后，前端点击取消按钮。   **监控指标：**   * 监控定点浇水任务是否终止。 * 检查前端是否显示定点浇水的取消反馈。 * 监控已经被浇花的植物含水量是否更新，未浇花的植物含水量是否保持不变。   **异常处理：**   * 如果定点浇水实时取消过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**   * 定点浇水任务终止，前端能看到定点浇水的取消反馈。 * 已经被浇花的植物更新含水量，未浇花的植物含水量不变。 |
| 测试成功标准 | 定点浇水的实时取消功能正常。 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中温室主要通道上无障碍，花盆没有消失，水泵含水量充足。  机器人处于等待(pending)状态 |

**（4）黑盒测试用例4**

**表135 定点浇水测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Target-04 |
| 优先级 | P0 |
| 测试类别 | 功能测试（导航异常处理） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中花盆没有消失，水泵含水量充足。 * 温室主要通道上存在障碍，机器人与目标花盆不可达。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 通过放置障碍物使得机器人与某一个目标花盆A之间不存在路径。 * 前端点击定点浇水按钮。 * 前端选择待浇水的一组花盆，其中包含目标花盆A。   **监控指标：**   * 监控机器人是否正常进行定点浇水。 * 当目标为目标花盆A时，检查前端是否收到浇花失败的反馈和导航路径规划失败的反馈。 * 监控机器人是否继续对剩余目标花盆浇水。 * 检查除目标花盆A的含水量不变外，其余目标花盆含水量是否更新。   **异常处理：**   * 如果处理导航异常过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**   * 机器人正常定点浇水。 * 当目标为目标花盆A时，前端收到浇花失败的反馈和导航路径规划失败的反馈，并继续对剩余目标花盆浇水。 * 除目标花盆A的含水量不变外，其余目标花盆含水量更新。 |
| 测试成功标准 | 机器人定点浇水中处理导航异常的功能正常。 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中花盆没有消失，水泵含水量充足。  温室主要通道上存在障碍，机器人与目标花盆不可达。  机器人处于等待(pending)状态 |

**（5）黑盒测试用例5**

**表136 定点浇水测试黑盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Target-05 |
| 优先级 | P0 |
| 测试类别 | 功能测试（浇水模块异常处理） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中温室主要通道上无障碍，花盆没有消失。 * 确保水泵含水量不足以完成此次定点浇水任务。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击定点浇水按钮。 * 前端选择足够数量的一组待浇水的花盆，使得任务过程中水泵含水量不足。   **监控指标：**   * 监控机器人是否正常进行定点浇水。 * 当水泵含水量为0时，检查前端是否收到浇花失败的反馈和水泵缺水的反馈。 * 监控机器人是否自动停止定点浇水任务。 * 检查因任务终止导致的未浇水的花盆的含水量是否不变，已浇水的花盆含水量是否更新。   **异常处理：**   * 如果处理浇水模块异常过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**   * 当水泵含水量为0时，前端收到浇花失败的反馈和水泵缺水的反馈，并自动停止定点浇水任务。 * 因任务终止导致的未浇水的花盆的含水量不变，其余植物含水量更新。 |
| 测试成功标准 | 机器人定点浇水中处理浇水模块异常的功能正常。 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中温室主要通道上无障碍，花盆没有消失。  水泵含水量不足以完成当此定点浇水任务。  机器人处于等待(pending)状态 |

#### 自动巡检浇水测试

**表137 自动巡检浇水测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-SYS-AutoWater |
| 相关需求 | 自动巡检浇水需求 |
| 测试涉及对象 | 前端，core，auto\_water，navigation，object\_detect，jet |

**（1）黑盒测试用例1**

**表138 自动巡检浇水测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-AutoWater-01 |
| 优先级 | P0 |
| 测试类别 | 功能测试 |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 确保已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中温室主要通道上无障碍，花盆没有消失，水泵含水量充足。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击自动巡检浇水按钮。   **监控指标：**   * 监控机器人是否导航至缺水花盆前方，并通过局部视觉校正将喷头对准花盆并浇水。 * 检查前端是否显示浇花进度反馈。 * 检查所有缺水花盆的含水量是否更新。   **异常处理：**   * 如果自动巡检浇水过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**   * 机器人导航至缺水花盆前方，并通过局部视觉校正将喷头对准花盆，并浇水。重复该过程至浇完最后一盆缺水的花。 * 在过程中能在前端看到浇花进度反馈。 * 所有缺水花盆含水量更新。 |
| 测试成功标准 | 机器人自动巡检浇水基本功能正常 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中温室主要通道上无障碍，花盆没有消失，水泵含水量充足。  机器人处于等待(pending)状态 |

**（2）黑盒测试用例2**

**表139 自动巡检浇水测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-AutoWater-02 |
| 优先级 | P0 |
| 测试类别 | 功能测试（异常处理） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中温室主要通道上无障碍，但缺水的目标花盆A被移除，水泵含水量充足。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 在建图后搬走一盆花A，且A缺水。 * 前端点击自动巡检浇水按钮。   **监控指标：**   * 监控机器人导航至被移除的花盆A的位置时是否抛出花盆消失的异常。 * 检查前端是否显示该异常反馈。 * 监控机器人抛出异常后是否继续正常进行巡航浇花。 * 检查除花盆A外的其余目标花盆含水量是否更新。   **异常处理：**   * 如果处理花盆消失异常过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**   * 当机器人导航至被移除的花盆A的位置时，抛出花盆消失的异常，前端能看到该异常反馈。 * 抛出异常后继续正常进行巡航浇花。 * 除花盆A的含水量不变，其余目标花盆含水量更新。 |
| 测试成功标准 | 机器人自动巡检浇水的处理花盆消失异常的功能正常。 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中温室主要通道上无障碍，但缺水的目标花盆A被移除，水泵含水量充足。  机器人处于等待(pending)状态 |

**（3）黑盒测试用例3**

**表140 自动巡检浇水测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-AutoWater-03 |
| 优先级 | P0 |
| 测试类别 | 功能测试（实时取消） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中温室主要通道上无障碍，花盆没有消失，水泵含水量充足。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击自动巡检浇水按钮。 * 在自动巡检浇水任务完成前，前端点击取消按钮。   **监控指标：**   * 监控自动巡检浇水任务是否终止。 * 检查前端是否显示自动巡检浇水的取消反馈。 * 检查已经被浇花的植物含水量是否更新，未浇花的植物含水量是否不变。   **异常处理：**   * 如果实时取消过程中出现异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**   * 自动巡检浇水任务终止，前端能看到自动巡检浇水的取消反馈。 * 已经被浇花的植物更新含水量，未浇花的植物含水量不变。 |
| 测试成功标准 | 自动巡检浇水的实时取消功能正常。 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中温室主要通道上无障碍，花盆没有消失，水泵含水量充足。  机器人处于等待(pending)状态 |

**（4）黑盒测试用例4**

**表141 自动巡检浇水测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-AutoWater-04 |
| 优先级 | P0 |
| 测试类别 | 功能测试（导航异常处理） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中花盆没有消失，水泵含水量充足。 * 温室主要通道上存在障碍，机器人与目标花盆不可达。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 通过放置障碍物使得机器人与某一个缺水目标花盆A之间不存在路径。 * 前端点击自动巡检浇水按钮。   **监控指标：**   * 监控机器人是否正常自动巡检浇水。 * 检查前端是否在目标为缺水目标花盆A时收到浇花失败的反馈和导航路径规划失败的反馈。 * 检查机器人是否继续对剩余目标花盆浇水。 * 检查除目标花盆A外的其余目标花盆含水量是否更新。   **异常处理：**   * 如果处理导航异常过程中出现其他异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**   * 机器人正常自动巡检浇水。 * 当目标为缺水目标花盆A时，前端收到浇花失败的反馈和导航路径规划失败的反馈，并继续对剩余目标花盆浇水。 * 除目标花盆A的含水量不变，其余目标花盆含水量更新。 |
| 测试成功标准 | 机器人自动巡检浇水中处理导航异常的功能正常。 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中花盆没有消失，水泵含水量充足。  温室主要通道上存在障碍，机器人与目标花盆不可达。  机器人处于等待(pending)状态 |

**（5）黑盒测试用例5**

**表142 自动巡检浇水测试黑盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-AutoWater-05 |
| 优先级 | P0 |
| 测试类别 | 功能测试（浇水模块异常处理） |
| 测试流程 | **准备阶段：**   * 启动系统，确保所有模块正常工作。 * 已经保存了一张地图，花盆含水量已经初始化。 * 确保浇花过程中温室主要通道上无障碍，花盆没有消失。 * 确保水泵含水量不足以完成当前自动巡检浇水任务。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击自动巡检浇水按钮，并且此时缺水花盆数量足够多使得任务过程中水泵含水量不足。   **监控指标：**   * 监控机器人是否正常执行自动巡检浇水任务。 * 检查前端是否在水泵含水量为0时收到浇花失败的反馈和水泵缺水的反馈。 * 检查自动巡检浇水任务是否自动停止。   **异常处理：**   * 如果处理浇水模块异常过程中出现其他异常，中止操作并记录异常信息。 * 重新启动系统并重复测试步骤。 |
| 测试结果 | **预期结果：**  当水泵含水量为0时，前端收到浇花失败的反馈和水泵缺水的反馈，并自动停止自动巡检浇水任务。 |
| 测试成功标准 | 机器人自动巡检浇水中处理浇水模块异常的功能正常。 |
| 用例备注 | 已经保存了一张地图，花盆含水量已经初始化。  浇花过程中温室主要通道上无障碍，花盆没有消失。  水泵含水量不足以完成当此自动巡检浇水任务。  机器人处于等待(pending)状态 |

#### 等待服务测试

**表143 等待服务测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-SYS-Pending |
| 相关需求 | 等待服务需求 |
| 测试涉及对象 | 前端，core，pending，map\_provider，navigation |

**（1）黑盒测试用例1**

**表144 等待服务测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Pending-01 |
| 优先级 | P0 |
| 测试类别 | 功能测试 |
| 测试流程 | **准备阶段：**   * 确保机器人处于温室中，并保存了一张地图。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击手动设置位置按钮。 * 在地图页面上选择一个点作为机器人的初始位置。   **监控指标：**   * 前端显示的初始位置是否更新。   **异常处理：**   * 如果机器人初始位置未更新，记录错误信息并重试。 |
| 测试步骤 | 同测试输入 |
| 测试结果 | **预期结果：**  前端显示的机器人初始位置变为用户手动设置的位置。 |
| 测试成功标准 | 机器人等待状态的手动设置位置功能正常。 |
| 用例备注 | 已经保存了一张地图，机器人处于温室中  机器人处于等待(pending)状态 |

**（2）黑盒测试用例2**

**表145 等待服务测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Pending-02 |
| 优先级 | P0 |
| 测试类别 | 功能测试 |
| 测试流程 | **准备阶段：**   * 确保机器人处于温室中，并保存了一张地图。 * 确保机器人处于等待(pending)状态。 * 确保机器人位置与目标航点之间的最短路径上不存在新增障碍。   **执行步骤：**   * 前端点击导航按钮。 * 前端选择航点。   **监控指标：**   * 机器人是否导航至选定的航点。 * 前端是否显示导航进度反馈。   **异常处理：**  如果机器人未能导航至选定的航点，记录错误信息并重试。 |
| 测试结果 | **预期结果：**  机器人导航至选定的航点，并且前端能看到导航进度反馈 |
| 测试成功标准 | 机器人等待状态的导航服务功能正常。 |
| 用例备注 | 已经保存了一张地图，机器人处于温室中  机器人处于等待(pending)状态。  机器人位置与目标航点之前的最短路径上不存在新增障碍。 |

**（3）黑盒测试用例3**

**表146 等待服务测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Pending-03 |
| 优先级 | P0 |
| 测试类别 | 功能测试（导航避障） |
| 测试流程 | **准备阶段：**   * 确保机器人处于温室中，并保存了一张地图。 * 确保机器人处于等待(pending)状态。 * 确保机器人位置与目标航点之间存在多条路径，其中最短路径上存在新增障碍。   **执行步骤：**   * 前端点击导航按钮。 * 前端选择航点。 * 在路径上放置障碍物，并保证存在另一条路径能够到达航点。   **监控指标：**   * 机器人是否在遇到障碍物时重新计算路径。 * 机器人是否绕开障碍物并最终到达航点。   **异常处理：**   * 如果机器人未能重新计算路径或未能绕开障碍物，记录错误信息并重试。 |
| 测试结果 | **预期结果**：  机器人导航至障碍物处时迅速重新计算路径，绕开障碍物，最终到达航点。 |
| 测试成功标准 | 机器人等待状态的导航避障功能正常。 |
| 用例备注 | 已经保存了一张地图，机器人处于温室中  机器人处于等待(pending)状态  机器人位置与目标航点之前的存在多条路径，其中最短路径上存在新增障碍。 |

**（4）黑盒测试用例4**

**表147 等待服务测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Pending-04 |
| 优先级 | P0 |
| 测试类别 | 功能测试（导航路径规划异常处理） |
| 测试流程 | **准备阶段：**   * 确保机器人处于温室中，并保存了一张地图。 * 确保机器人处于等待(pending)状态。确保机器人位置与目标航点之间的所有路径上都存在障碍。   **执行步骤**：   * 前端点击导航按钮。 * 前端选择航点。 * 在机器人和航点的所有路径上都放置障碍物。   **监控指标：**   * 机器人是否尝试搜索所有路径。 * 前端是否显示导航失败的反馈。   **异常处理：**   * 如果机器人未能正确处理路径规划失败或前端未显示导航失败的反馈，记录错误信息并重试。 |
| 测试结果 | **预期输出：**  机器人导航，尝试搜索所有路径后，最终发现航点不可达。前端看到导航失败的反馈 |
| 测试成功标准 | 机器人等待状态的导航路径规划失败的异常处理功能正常。 |
| 用例备注 | 已经保存了一张地图，机器人处于温室中  机器人处于等待(pending)状态  机器人位置与目标航点之前的所有路径上都存在障碍。 |

**（5）黑盒测试用例5**

**表148 等待服务测试黑盒测试用例5说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-Pending-05 |
| 优先级 | P0 |
| 测试类别 | 功能测试（导航实时取消） |
| 测试流程 | **准备阶段：**   * 确保机器人处于温室中，并保存了一张地图。 * 确保机器人处于等待(pending)状态。   **执行步骤：**   * 前端点击导航按钮。 * 前端选择航点。 * 在到达之前点击取消按钮。   **监控指标：**  机器人是否在点击取消按钮后迅速停止移动。  **异常处理：**  如果机器人未能在点击取消按钮后迅速停止移动，记录错误信息并重试。 |
| 测试结果 | **预期结果：**  点击取消按钮后机器人迅速停止移动。 |
| 测试成功标准 | 机器人等待状态的导航实时取消功能正常。 |
| 用例备注 | 已保存了一张地图，机器人处于温室中  机器人处于等待(pending)状态 |

#### 系统性能测试

**表149 性能测试用例总说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试项目标识 | WJ-SYS-PERF |
| 相关需求 | 各业务需求 |
| 测试涉及对象 | 机器人总体硬件模块与软件功能包 |
| 前置条件 | 条件1: 机器人启动且工作正常  条件2:前端用户界面与ROS端均已经正常启动 |

**（1）黑盒测试用例1**

**表150 系统性能测试黑盒测试用例1说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-PERF-01 |
| 优先级 | P0 |
| 测试类别 | 性能测试 |
| 测试流程 | * 用手柄控制机器人向前移动约1米。 * 使用计时工具测量从发送指令到机器人开始移动的时间，同时记录该时间。 * 重复以上步骤10次，记录每次的响应时间。 |
| 测试结果 | 记录的响应时间。 |
| 测试成功标准 | 响应时间平均值应小于200毫秒，且最大响应时间不超过250毫秒。 |
| 用例备注 | 测试环境应尽量保持一致，避免外界干扰，测试实时性和响应时间。 |

**（2）黑盒测试用例2**

**表151 系统性能测试黑盒测试用例2说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-PERF-02 |
| 优先级 | P0 |
| 测试类别 | 性能测试 |
| 测试流程 | * 启动ROS机器人系统，确保所有节点正常运行。 * 使用系统监控工具记录CPU和内存使用情况。 * 在高负载条件下运行机器人（如在执行多个花盆浇水任务的时候）。 * 持续监控并记录CPU和内存使用情况。 |
| 测试结果 | 记录的CPU和内存使用率。 |
| 测试成功标准 | CPU使用率应低于90%，内存使用率应低于90%。 |
| 用例备注 | 测试应在多次不同任务场景下进行，以确保数据的全面性，测试机器人的资源利用率。 |

**（3）黑盒测试用例3**

**表152 系统性能测试黑盒测试用例3说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-PERF-03 |
| 优先级 | P0 |
| 测试类别 | 性能测试 |
| 测试流程 | * 启动ROS机器人系统，连续运行1小时左右。 * 记录系统运行过程中是否出现崩溃、卡顿或其他异常。 * 定期记录系统的运行状态（如每5分钟记录一次）。 * 测试结束后，分析所有记录的数据。 |
| 测试结果 | 记录的系统状态和异常情况。 |
| 测试成功标准 | 系统在1小时内无崩溃或严重异常 |
| 用例备注 | 测试浇花机器人的稳定性和可靠性 |

**（4）黑盒测试用例4**

**表153 系统性能测试黑盒测试用例4说明表**

|  |  |
| --- | --- |
| **测试用例项** | **描述** |
| 测试用例编号 | WJ-SYS-PERF-04 |
| 优先级 | P0 |
| 测试类别 | 性能测试 |
| 测试流程 | * 完全充满机器人电池，确保机器人初始状态是满电。 * 启动ROS机器人系统，让其在高负载任务下（执行建图、导航、浇花等业务需求）连续运行。 * 每隔5分钟记录电池剩余电量和系统的电源管理行为。 * 记录机器人在高负载任务下的续航时间，直到电量耗尽或低于安全运行阈值。 * 测试结束后，分析所有记录的数据。 |
| 测试结果 | 记录的电池剩余电量、电源管理行为和续航时间。 |
| 测试成功标准 | 机器人应能连续运行至少1小时，且电源管理功能正常（如在低电量时自动切换到节电模式）。 |
| 用例备注 | 测试浇花机器人的电池和电源管理系统以及续航能力。 |

# 测试进度表

## 测试时间安排

**表154 测试时间安排**

|  |  |
| --- | --- |
| **预计测试时间** | **测试范围** |
| 2024年5月20日前 | 机器人硬件部分   * WJ-ROBOT-MV * WJ-ROBOT-JS * WJ-ROBOT-RADAR * WJ-ROBOT-CAMERA * WJ-ROBOT-SPEAKER * WJ-ROBOT-ARM * WJ-ROBOT-PS   ROS 端部分   * WJ-ROS-CT * WJ-ROS-NAV * WJ-ROS-MP * WJ-ROS-OD * WJ-ROS-RA * WJ-ROS-DB |
| 2024年5月24日前 | 前端单元功能   * WJ-FRONTEND-APPBAR * WJ-FRONTEND-POTLIST * WJ-FRONTEND-OPPANEL * WJ-FRONTEND-FOOTER   集成测试   * WJ-INT-MapInit * WJ-INT-Navigation * WJ-INT-WaterJet * WJ-INT-ServiceInit   系统测试   * WJ-SYS-AutoMap * WJ-SYS-ManualMap * WJ-SYS-Inspection * WJ-SYS-Target * WJ-SYS-AutoWater * WJ-SYS-Pending |

## 进度约束

### 测试阶段进度约束

本软件测试计划(STP)确立了三个核心测试阶段：单元测试、集成测试及系统测试，旨在系统性地验证浇花机器人的系统质量。单元测试注重于验证最小可测试单元的功能正确性，为后续阶段奠定基础。集成测试旨在检验各模块间的协同工作能力及其功能整合的准确性，其依赖于单元测试能够成功完成测试。最后，系统测试对业务需求进行全面覆盖，不仅验证整体功能的完整性，还确保软件在业务场景中运行的准确无误，此阶段依赖于单元测试与集成测试均已测试完成。

本软件测试计划的测试阶段约束清晰展现了测试活动由局部到整体的递进逻辑和相互依存关系，从而确保了测试的全面性。

### 测试样例进度约束

在测试阶段中，具体进度约束的表格如下：

**表155 测试时间安排**

|  |  |
| --- | --- |
| **测试样例** | **前置测试样例** |
| ROBOT-JS | ROBOT-MV |
| ROBOT-RADAR | ROBOT-MV |
| ROBOT-CAMERA | ROBOT-MV |
| ROBOT-PS | ROBOT-JS |
| ROBOT-RADAR |
| ROBOT-CAMERA |
| ROBOT-SPEAKER |
| ROBOT-ARM |
| ROS-OD | ROS-PS |
| ROS-NAV | ROS-PS |
| ROS-RA | ROS-PS |
| ROS-MP | ROS-NAV |
| FRONTEND-OPPANEL | ROS-OD |
| ROS-MP |
| ROS-RA |
| ROS-CT |
| ROS-DB |
| FRONTEND-MAP | ROS-MP |
| ROS-DB |
| FRONTEND-APPBAR | ROS-CT |
| FRONTEND-FOOTER | ROS-CT |
| FRONTEND-POTLIST | ROS-DB |
| INT-MapInit | FRONTEND-OPPANEL |
| FRONTEND-MAP |
| INT-Navigation | FRONTEND-OPPANEL |
| FRONTEND-MAP |
| INT-Water | FRONTEND-OPPANEL |
| FRONTEND-MAP |
| SYS-AutoMap | INT-MapInit |
| SYS-ManualMap | INT-MapInit |
| SYS-Inspection | INT-Navigation |
| SYS-Target | INT-Navigation |
| INT-Water |
| SYS-AutoWater | INT-Navigation |
| INT-Water |
| SYS-Pending | INT-ServiceInit |

### 关键路径法分析

在本节中，我们将采用关键路径法（Critical Path Method, CPM）来识别最长的连续活动序列，确保这些活动按时完成以满足最终期限。CPM连续活动序列图如下图所示，其中数字代表预计测试用例所需要的测试时间（以小时为单位）。

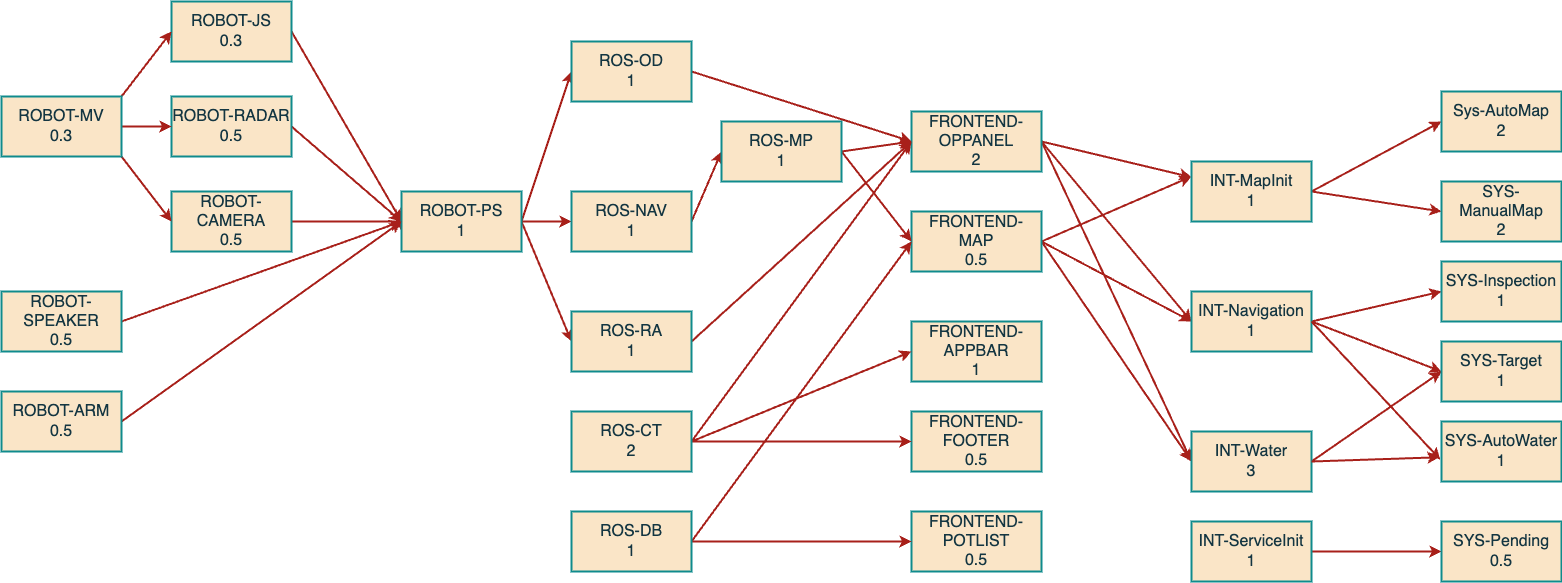


图 5.1 CPM分析方法可视化

根据关键路径法分析可以得知最耗时的路径为：

* ROBOT-MV 、（ROBOT-RADAR或ROBOTJS）、ROBOT-PS、ROS-NAV、ROS-MP、FRONTEND-OPPANEL、INT-Water、（SYS-Target或SYS-AutoWater）

在执行测试的时候，我们将着重关注最耗时的路径从而确保整个项目的测试计划能够按照计划向前推进。

# 各需求对应的测试

## 数据需求

**表156 数据需求对应的测试项**

|  |  |
| --- | --- |
| 需求名称 | 对应测试项 |
| 植物数据 | 植物数据库相关的测试 |
| 任务数据 | 包括所有从前端发送请求的测试，以及ROS端控制器接受任务的测试 |

## 功能需求

**表157 功能需求对应的测试项**

|  |  |
| --- | --- |
| 需求名称 | 对应测试项 |
| 导航与动态避障 | WJ-ROS-NAV, WJ-INT-Navigation |
| 自动建图与手动建图 | WJ-ROS-MP, WJ-INT-MapInit |
| 花盆识别 | WJ-ROS-OD |
| 浇水 | WJ-ROS-RA, WJ-INT-WaterJet |
| 数据库服务 | WJ-ROS-DB |
| 用户页面 | WJ-FRONTEND |
| 系统急停 | WJ-ROS-CT-09 |

## 异常处理需求

**表158 异常处理需求对应的测试项**

|  |  |
| --- | --- |
| 需求名称 | 对应测试用例 |
| 航点不可达异常 | WJ-ROS-NAV-02, WJ-SYS-Target-04, WJ-SYS-AutoWater-04, WJ-SYS-Pending-04 |
| 建图不完全异常 | WJ-ROS-MP-04, WJ-INT-MapInit-04, WJ-SYS-AutoMap-03 |
| 花盆缺失异常 | WJ-ROS-RA-03, WJ-INT-WaterJet-04, WJ-SYS-Target-02, WJ-SYS-AutoWater-02 |
| 水泵缺水异常 | WJ-ROS-RA-02, WJ-SYS-Target-05, WJ-SYS-AutoWater-05 |
| 系统启动异常 | WJ-INT-ServiceInit-02 |

## 业务需求

**表159 业务需求对应的测试项**

|  |  |
| --- | --- |
| 需求名称 | 对应测试项 |
| 自动建图 | WJ-SYS-AutoMap |
| 手动建图 | WJ-SYS-ManualMap |
| 巡检维护 | WJ-SYS-Inspection |
| 定点浇水 | WJ-SYS-Target |
| 自动巡检浇水 | WJ-SYS-AutoWater |
| 等待服务 | WJ-SYS-Pending |

## 非功能需求

**表160 非功能需求对应的测试项**

|  |  |
| --- | --- |
| 需求名称 | 对应测试项 |
| 响应时间 | 机器人单元测试，前端单元测试，系统测试中关于响应速度的测试。 |
| CPU负载 | 系统测试中关于CPU负载的测试。 |
| 电源系统 | 系统测试中关于电源系统的测试。 |

# 评价

## 评价准则

在评价软件测试计划时，我们将基于以下准则来衡量其质量与有效性：

* 完整性：评估软件测试计划是否全面覆盖了项目定义的所有需求，包括数据需求、功能需求、异常处理需求、业务需求和非功能需求；
* 合理性：分析测试策略与方法是否切合实际，如测试级别的划分、测试类别设定是否恰当，以及测试环境的选择是否满足真实场景的模拟；
* 可执行性：检验测试计划的可行性，包括测试用例的详细程度、资源分配、时间表的合理性以及人员职责的明确性；
* 可追溯性：考察需求与测试用例之间的关联性，确保每个需求都能通过至少一个测试用例得到验证；
* 持续改进机制：评估计划中是否包含了反馈循环，如复盘会议、文档修订和测试结果的记录与分析，以驱动持续的优化。

## 数据处理

数据处理是确保测试有效性与决策质量的重要环节，即便当前阶段尚无实际测试结果，预设一套高效的数据管理流程对于后续的测试分析与优化至关重要。在数据处理部分中，我们的软件测试计划文档具有以下特点：

* 标准化模板: 设计并采用标准化的数据收集模板，确保所有测试人员遵循一致的格式记录测试数据，包括但不限于测试用例ID、测试环境、测试步骤、既定的预期结果、实际观测的结果、执行日期及时间、测试人员等信息。标准化模板可有效确保每位测试者遵循统一的数据记录规范，提升团队协作效率和后续分析的可靠性。
* 基于数据的决策: 我们计划将测试样例中的分析结果提炼成为具体可行的改进建议，指导后续测试策略调整等决策。这些数据将直接指导测试策略的适时调整和潜在缺陷的优先级排序，确保测试活动能够高效覆盖项目的各个方面，推动产品质量的持续提升。
* 异常测试样例识别：我们计划对异常数据进行单独审查，判断其对整体结论的影响，必要时进行剔除或特别标记。该举措有助于我们保持数据分析的准确性，同时也为更深层次的问题根源分析保留了线索。

## 结论

基于以上准则，对当前软件测试计划的前瞻性评价如下：

本软件测试计划在理论上构建了一个详尽而系统的测试框架，覆盖了从单元测试到系统测试的多个层面，确保了软件质量的全方位检验。它明确界定了测试环境，并为各类测试提供了详尽的记录规范，这有利于提高测试的透明度和可追溯性。此外，测试进度表和需求的可追踪性设计，展示了项目管理上的周到考虑，有助于团队协同与项目进度的有序进行。

实际执行效果还需在测试周期中通过测试后的复盘会议、测试报告与数据分析来验证。软件测试计划的灵活性与适应性也是未来评估的一个关键点，为未来实际测试在面对未预见的问题或浇花机器人需求变更时，软件测试计划能够快速迭代与调整，从而适合当前最新的需求。

综上，本软件测试计划构建了一套坚实的基础，为即将开展的测试活动提供了明确的指南。其成功与否，将在后续的执行阶段通过实施效果、问题发现与解决效率，以及软件最终的质量表现来验证。