**浇花机器人——温室管家**

**软件开发计划**

**SDP-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** |
| 9 | LTT | 主编, 整合文档, 过程模型 |
| 6 | Kun | 评审, 资源计划, 进度计划 |
| 2 | Hao7un | 范围 |
| 1 | Le | 项目任务概要 |
| 5 | Mxode | 风险管理 |

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **提交日期** | **主要编制人** | **审核人** | **版本说明** |
| V1.0 | 2024.3.15 | LTT | Kun | 软件开发计划初稿 |
| V1.1 | 2024.4.10 | LTT | Kun | 更新进度计划 |
| V1.2 | 2024.5.1 | LTT | Kun | 更新进度计划 |
| V1.3 | 2024.6.9 | Le | LTT | 更新WBS图 |
| V2.0 | 2024.6.9 | LTT | Kun | 校对, 同步进度计划 |

# 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。

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，通过实现自动浇灌功能，从而降低人工成本并提高温室管理的智能化水平。

**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：

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

**非功能性需求：**

* 可靠性与稳定性：为了保证高效、安全地完成用户的需求，浇花机器人需要具有可靠性、稳定性等特性，为用户提供良好的使用体验。
* 异常处理功能：机器人拥有完备的异常处理功能，能够实时反馈任务进度，并在异常情况发生时自动采取恢复措施（或将问题上报给用户），确保系统能够稳定运行。

**应用场景：**

* **种植产业温室**：适用于不同规模的温室花卉种植，提供个性化的浇水方案。
* **科研机构**：为研究人员提供精准控制的植物水分条件，助力农业科研。
* **城市农业项目**：适用于城市屋顶花园、垂直农业等新型农业形态

## 文档概述

**用途：**

本软件开发计划的主要用途是为开发小组成员明确项目的组内分工、时间安排以及软件开发方向，帮助组内各个成员保持进度的一致性，从而顺利完成项目的开发。

**内容组织：**

* 项目任务概要
* 风险管理
* 过程模型
* 资源计划
* 进度计划

## 术语和缩略词

**术语：**

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

ROS Melodic： 机器人操作系统

C++: 一种支持面向对象和底层操作的高性能编程语言

Python：一种简洁易懂，功能强大的解释型高级编程语言

Rviz：机器人可视化开发平台

Gazebo：机器人仿真开发平台

Visual Studio Code：代码开发工具

GitLab：基于Git的集成软件开发平台

**缩略词：**

表1 缩略词/全称表

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

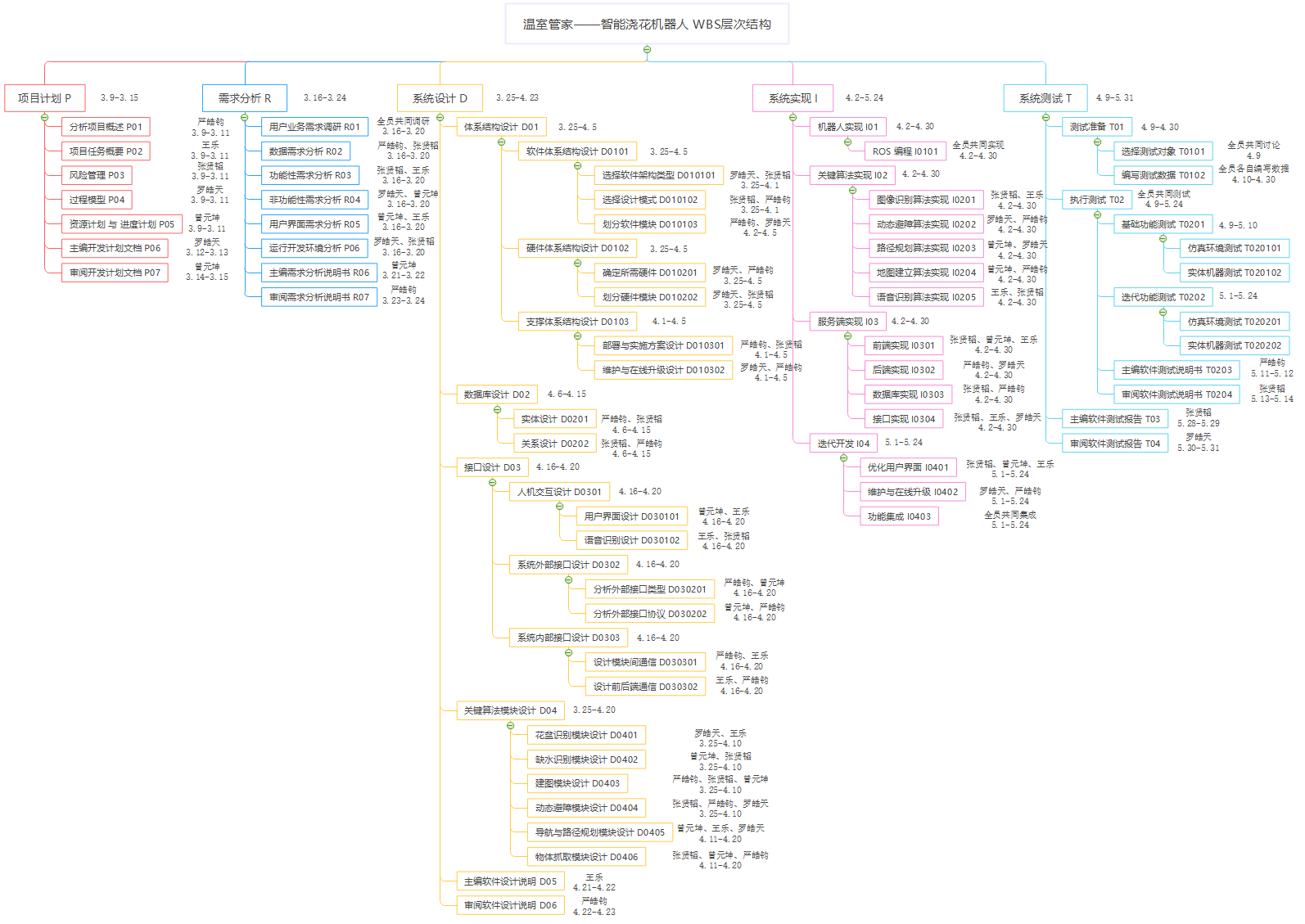
## 引用文档

[1] 北京六部工坊科技有限公司，《启智ROS版\_开发手册\_20181109》，2018-11.9

[2] 机械工业出版社，《ROS机器人开发实践》，2018-5.1

# 项目任务概要

## 工作内容



[图 1 WBS图](file:///D:\Documents\Edge%20Downloads\image\WBS.png)

## 主要人员

表2 开发人员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓名** | **角色** | **工作时间** | **工作经验** | **技术水平** |
| LTT | 项目组长  程序开发人员  部署人员  评审人员 | 全程 | 有前端开发经历, 熟悉前后端技术, 项目管理和部署 | 熟悉 C++, Python, Vue 技术 |
| Hao7un | 系统架构师  程序开发人员  测试人员  评审人员 | 全程 | 有小型团队项目开发经验，有Django后端开发经验 | 掌握C，Python，Java等语言，拥有Django开发能力 |
| Mxode | 系统分析员  程序开发人员 部署人员  评审人员 | 全程 | 有多个团队项目开发经验，掌握 Django、FastAPI 后端开发经验，熟悉前端开发 | 熟练使用 Python、Java，掌握 C/C++ 编程，熟悉 Linux 开发环境 |
| Kun | 程序开发人员  部署人员  评审人员 | 全程 | 有数据库团队项目开发经验，Vue前端开发经验，Rails敏捷开发经验 | 掌握Java，Python，C等编程语言 |
| Le | 程序开发人员  测试人员  评审人员 | 全程 | 拥有小型团队项目经验，具备前端开发能力 | 掌握Java、Python、C编程语言，熟悉Vue3框架 |

## 产品

### 程序、数据或设备

* 项目源代码
* 可执行程序
* 安装程序
* 数据文件
* 配置完毕的ROS启智机器人及其外围设备

### 文档

1. SDP-软件开发计划.doc
2. SRS-软件需求规格说明书.doc
3. SDD-软件设计文档.doc
4. STP-软件测试计划.doc
5. STR-软件测试报告.doc

## 运行与开发环境

### 运行环境

* 硬件环境：启智 ROS 机器人、机械臂等配套外围设备、供电与连接线缆
* 软件环境：Ubuntu 18.04 LTS、ROS Melodic

### 开发环境

* 硬件环境：安装 Ubuntu 18.04 LTS 的个人计算机、启智 ROS 机器人[1-2]及其配套外围设备、供电与连接线缆、开发板、继电器、水泵等浇花套件
* 软件环境：Ubuntu 18.04 LTS、ROS Melodic、代码编辑器 Visual Studio Code、可视化平台 Rviz 以及仿真平台 Gazebo、Python 2.7、项目管理平台 GitLab

## 项目期限

本项目从2024年3月9日开始，持续至2024年6月14日结束。

# 风险管理

## 风险识别

表3 风险识别表

|  |  |
| --- | --- |
| **风险类型** | **具体示例** |
| 估算风险  Estimation Risk | 错误估计项目总体开发时间 |
| 错误估计各任务持续时间 |
| 对成本存在估计偏差 |
| 依赖于过时的估算方法或工具 |
| 技术风险  Technical Risk | 新技术的稳定性和成熟度未知 |
| 集成测试和兼容性问题 |
| 第三方软件或硬件的更新或更改 |
| 安全漏洞或数据丢失风险 |
| 人员风险  People Risk | 团队成员缺乏必要的技能或经验 |
| 团队之间的沟通障碍 |
| 个人目标与项目目标之间不匹配 |
| 人员流失导致进度延迟或受损 |
| 需求风险  Requirement Risk | 需求不明确或变更频繁 |
| 需求文档或说明不清晰 |
| 需求变更管理缺乏流程或管控 |
| 用户验收标准不明确或过于严格 |
| 法律合规风险  Legal and Compliance Risk | 数据隐私或安全法规变更 |
| 对开源软件或许可问题的理解不足 |
| 数据或隐私政策的合规性问题 |
| 数据处理和存储的合规性问题 |

## 可能性分析与应对策略

表4 风险应对策略表

|  |  |  |
| --- | --- | --- |
| **风险** | **可能性** | **应对策略** |
| 错误估计项目总体开发时间 | 高 | 使用历史数据和专业意见来改进估算技术和方法 |
| 错误估计各任务持续时间 | 高 | 采用敏捷方法，定期评估任务完成情况，并根据反馈进行调整 |
| 对成本存在估计偏差 | 中 | 使用成本模型和敏感度分析来评估可能的偏差，并为不确定性做好准备 |
| 依赖于过时的估算方法或工具 | 中 | 采用最新的项目管理工具和技术，不断学习和改进估算方法 |
| 新技术的稳定性和成熟度未知 | 低 | 进行原型验证和实验，评估新技术的可行性和稳定性 |
| 集成测试和兼容性问题 | 高 | 制定全面的集成测试计划，并使用自动化测试工具进行测试 |
| 第三方软件或硬件的更新或更改 | 中 | 定期更新软件和硬件，并进行兼容性测试 |
| 安全漏洞或数据丢失风险 | 中 | 实施安全最佳实践，包括加密、权限管理等，定期进行安全漏洞扫描和修复 |
| 团队成员缺乏必要的技能或经验 | 低 | 团队技术栈互补，互相帮助，并积极学习不足的知识 |
| 团队之间的沟通障碍 | 中 | 定期召开组会，跟进成员任务进展，协调各成员之间的交流 |
| 个人目标与项目目标之间不匹配 | 中 | 与团队成员协商目标，确保他们理解和支持项目目标 |
| 人员流失导致进度受损或延迟 | 低 | 充分沟通，尽量激发队员积极性，并协调其余队员补充相应进度 |
| 需求不明确或变更频繁 | 高 | 实施变更管理流程，从用户侧出发，明确和稳定需求 |
| 需求文档或说明不清晰 | 中 | 定期审查需求文档，确保清晰理解和一致性 |
| 需求变更管理缺乏流程或管控 | 低 | 设立变更控制制度，严格审核和批准需求变更 |
| 用户验收标准不明确或过于严格 | 中 | 明确验收标准，确保符合实际需求 |
| 数据隐私或安全法规变更 | 低 | 定期审查法规变化，更新隐私政策和安全措施 |
| 对开源软件或许可的理解不足 | 中 | 进行开源软件审查，确保合规并了解其许可要求 |
| 数据或隐私政策的合规性问题 | 低 | 与专业人士交流，监督隐私政策和数据保护合规性 |
| 数据处理和存储的合规性问题 | 中 | 遵循相关法规，采取安全措施保护数据，并定期进行合规性审查 |

# 过程模型

项目采用渐增式增量迭代的开发过程模型。

## 开发准备工作(3月8日-3月24日)

1. 学习 ROS 相关知识, 搭建开发环境, 熟悉项目管理平台
   * 所有人, 3月8日-3月15日
2. 确定开发项目, 进行需求分析
   * 所有人, 3月8日-3月22日
3. 撰写软件需求规格说明书, 对需求文档评审, 导出
   * 需求文档主编(Kun), 审核人(Hao7un), 3月22日-3月24日

## 系统整体架构设计(3月25日-4月1日)

1. 调研相关系统实现, 设计系统整体架构, 梳理系统工作流程, 明确各个模块功能和接口规范
   * 所有人, 3月25日-3月30日
2. 学习相关技术, 评估开发难度
   * 所有人, 3月25日-4月1日
3. 撰写软件设计说明, 对设计文档评审, 导出
   * 设计文档主编(Hao7un), 审核人(Le), 3月31日-4月1日

## 系统框架基本实现(4月2日-4月30日)

1. 实现机器人系统基本框架和各个模块
   * 所有人, 4月2日-4月30日
2. 实现前端, ROS 端基本框架, 测试前端、前端-ROS端通信
   * 所有人, 4月9日-4月30日
3. 根据实际情况完善设计文档, 评审, 导出
   * 设计文档主编(Le), 审核人(Hao7un), 4月16日-4月23日
4. 撰写接口文档, 编写测试及测试文档
   * 所有人, 4月2日-4月30日

## 系统迭代开发(5月1日-5月24日)

1. 完善前端显示界面
   * LTT, 5月1日-5月24日
2. 迭代目标检测, 使用 YOLO 模型
   * Mxode, Hao7un, 5月1日-5月24日
3. 实现其他拓展功能
   * 所有人, 5月1日-5月24日
4. 完善测试及测试文档
   * 所有人, 5月1日-5月11日
5. 整合软件测试计划, 评审, 导出
   * 测试计划主编(Hao7un), 审核人(Mxode), 5月12日-5月14日
6. 撰写软件测试报告, 评审, 导出
   * 软件测试报告(Mxode), 审核人(LTT), 5月24日-5月31日

# 资源计划

软件资源需求如下：

1. Ubuntu 18.04 LTS
2. ROS Melodic
3. Visual Studio Code
4. Rviz可视化平台和Gazebo仿真环境
5. Git代码管理工具以及远程Gitlab
6. 启智机器人的模块和驱动代码

硬件需求如下：

1. PC机5台
2. 启智机器人
3. 启智机器人机械臂
4. 开发板、继电器、水泵、喷水软管

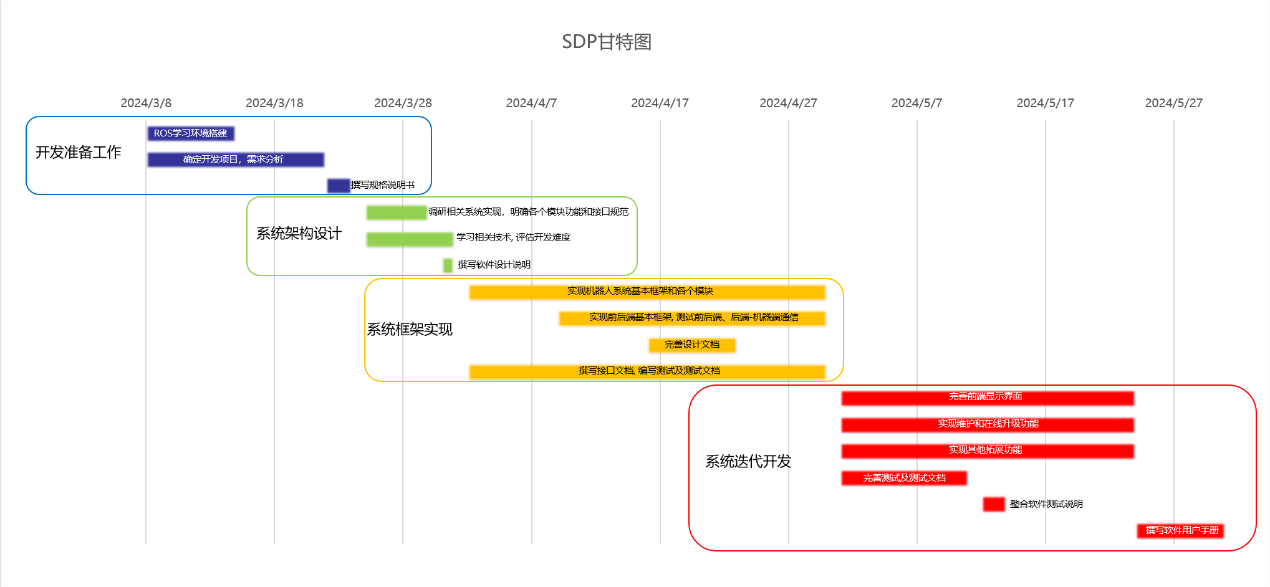
# 进度计划

表5 进度计划表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **活动名** | **开始** | **持续时间** | **结束** | **前置活动** |
| T01 | **开发准备工作** | 2024/3/8 | 16 | 2024/3/24 |  |
| T0101 | 1 ROS学习，环境搭建 | 2024/3/8 | 7 | 2024/3/15 |  |
| T0102 | 2 确定开发项目，需求分析 | 2024/3/8 | 14 | 2024/3/22 | T0101 |
| T0103 | 3 撰写规格说明书 | 2024/3/22 | 2 | 2024/3/24 | T0102 |
| T02 | **系统架构设计** | 2024/3/25 | 7 | 2024/4/1 | T01 |
| T0201 | 4 调研相关系统实现，明确各个模块功能和接口规范 | 2024/3/25 | 5 | 2024/3/30 |  |
| T0202 | 5 学习相关技术, 评估开发难度 | 2024/3/25 | 7 | 2024/4/1 |  |
| T0203 | 6 撰写软件设计说明 | 2024/3/31 | 1 | 2024/4/1 | T0201  T0202 |
| T03 | **系统框架实现** | 2024/4/2 | 28 | 2024/4/30 | T02 |
| T0301 | 7 实现机器人系统基本框架和各个模块 | 2024/4/2 | 28 | 2024/4/30 |  |
| T0302 | 8 实现前端, ROS 端基本框架, 测试前端、前端-ROS端通信 | 2024/4/9 | 21 | 2024/4/30 | T0301 |
| T0303 | 9 完善设计文档 | 2024/4/16 | 7 | 2024/4/23 |  |
| T0304 | 10 撰写接口文档, 编写测试及测试文档 | 2024/4/2 | 28 | 2024/4/30 |  |
| T04 | **系统迭代开发** | 2024/5/1 | 30 | 2024/5/31 | T03 |
| T0401 | 11 完善前端显示界面 | 2024/5/1 | 23 | 2024/5/24 |  |
| T0402 | 12 迭代目标检测, 使用 YOLO 模型 | 2024/5/1 | 23 | 2024/5/24 |  |
| T0403 | 13 实现其他拓展功能 | 2024/5/1 | 23 | 2024/5/24 |  |
| T0404 | 14 完善测试及测试计划 | 2024/5/1 | 10 | 2024/5/11 |  |
| T0405 | 15 整合测试计划 | 2024/5/12 | 2 | 2024/5/14 | T0404 |
| T0406 | 16撰写软件测试报告 | 2024/5/24 | 7 | 2024/5/31 | T0405 |

项目重要里程碑：

1. 开发准备工作结束：2023.3.24
2. 系统架构设计结束：2023.4.1
3. 系统框架实现结束：2023.4.30
4. 系统迭代开发：2023.5.31

[](file:///D:\Documents\Edge%20Downloads\image\SDP-进度计划甘特图.png)

[图2 SDP 进度计划甘特图](file:///D:\Documents\Edge%20Downloads\image\SDP-进度计划甘特图.png)