**简易机器人**

**开发计划**

***SDP 101组***

**V0.2.0**

分工说明

|  |  |  |
| --- | --- | --- |
| 小组名称 |  | |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 17373125 | 潘林煜 | 负责风险评估分析和主编全文 |
| 17373091 | 孙一丹 | 负责制定项目任务计划和确定里程碑节点 |
| 17373001 | 郑 奕 | 负责明确文档需求和撰写项目任务概要 |
| 17373003 | 施雨萌 | 负责建立过程模型和全文审查 |
| 17373355 | 郭若熙 | 负责制定范围章节和资源计划 |

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V0.1.0 | 2020.3.9 | 潘林煜等5人 | 施雨萌 | 项目开发计划第一版 |
| V0.2.0 | 2020.3.21 | 潘林煜等5人 | 施雨萌 | 项目开发计划第二版，增加甘特图 |
| V0.3.0 | 2020.6.7 | 潘林煜等5人 | 施雨萌 | 冻结文档前格式校对 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 范围 1](#_Toc34431757)

[1.1 项目概述 1](#_Toc34431758)

[1.2 文档概述 1](#_Toc34431759)

[1.3 术语和缩略词 2](#_Toc34431760)

[1.4 引用文档 3](#_Toc34431761)

[2. 项目任务概要 3](#_Toc34431762)

[2.1 工作内容 3](#_Toc34431763)

[2.2 主要人员 3](#_Toc34431764)

[2.3 产品 4](#_Toc34431765)

[2.3.1 程序、数据或设备 4](#_Toc34431766)

[2.3.2 文档 4](#_Toc34431767)

[2.4 运行与开发环境 5](#_Toc34431768)

[2.4.1 运行环境 5](#_Toc34431769)

[2.4.2 开发环境 6](#_Toc34431770)

[2.5 项目期限 6](#_Toc34431771)

[3. 风险管理 6](#_Toc34431772)

[4. 过程模型 8](#_Toc34431773)

[5. 资源计划 8](#_Toc34431774)

[6. 进度计划 9](#_Toc34431775)

[6.1 里程碑事件 9](#_Toc34431776)

[6.2 里程碑事件人员分配 10](#_Toc34431777)

# 范围

## 项目概述

本项目为北京航空航天大学计算机学院2020年春季学期《软件工程》课程开发项目。在课程组提供的ROS机器人等软硬件基础资源上开发一款实现避障、路径规划和检测抓取目标的简易机器人。机器人主要功能有主动控制、动态障碍物避障、实时建立环境地图、动态路径规划以及识别抓取目标物。进一步拓展可添加语音交互、巡检路径规划以及多目标检测等功能。

本项目实现的简易机器人应用广泛。避障功能的实现可以将简易机器人应用为广告机器人、前台接待机器人等简单机器人。路径规划功能的实现可以将其应用为扫地机器人、巡查机器人、领位机器人等需要检测环境的综合机器人。检测及抓取目标功能的实现可以将其应用为服务机器人、分拣搬运机器人等复杂的实用型机器人。

## 文档概述

本文档的用途是对项目开发计划进行分解规划、安排人员分工以及确定项目时间节点。文档内容分为范围、项目任务概要、风险管理、过程模型、资源计划和进度计划六个部分。本小组将按这六个部分学习并实现工程化的软件开发。

## 术语和缩略词

表1.3-1 术语与缩略词

|  |  |
| --- | --- |
| 术语或缩略词 | 解释或全称 |
| ROS | ROS（Robot Operating System）是一个适用于机器人的开源的元操作系统。它提供了操作系统应有的服务，包括硬件抽象，底层设备控制，常用函数的实现，进程间消息传递，以及包管理。它也提供用于获取、编译、编写、和跨计算机运行代码所需的工具和库函数。 |
| WBS | WBS工作分解结构（Work Breakdown Structure）， 创建WBS是把项目工作按阶段可交付成果分解成较小的，更易于管理的组成部分的过程。WBS是一个描述思路的规划和设计工具。它帮助项目经理和项目团队确定和有效地管理项目的工作。 |
| CCB | CCB(Change Control Board) 在CMMI(Capability Maturity Model Integration)中，是“变更控制委员会”的含义，同时具有配置控制委员会（Configuration Control Board）的含义。CCB可以由一个小组担任，也可以由多个不同的组担任，负责做出决定究竟将哪些已建议需求变更或新产品特性付诸应用。 |
| Ubuntu | Ubuntu是一个以桌面应用为主的Linux操作系统。 |

## 引用文档

[1]启智ROS机器人开发手册V1.1.0

# 项目任务概要

## 工作内容

图2.1-1 项目WBS图

## 主要人员

表2.2-1 人员及分工

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 开发人员 | 承担角色 | 工作时间 | 工作经验 | 技术水平 |
| 孙一丹 | 组长  系统分析员  程序开发 | 不少于480小时 | 极强的算法实现能力 | 精通 |
| 郑奕 | 系统架构师  程序开发 | 不少于480小时 | 一定规模系统开发经验 | 精通 |
| 施雨萌 | 程序开发  文档评审 | 不少于480小时 | 丰富的程序开发经验 | 精通 |
| 郭若熙 | 程序开发  测试人员 | 不少于480小时 | 逻辑思维与验证能力 | 精通 |
| 潘林煜 | 数据库架构师  文档评审 | 不少于480小时 | 一定程度的系统开发经验 | 熟练 |

## 产品

### 程序、数据或设备

1. 项目源代码

1.1 运动自主控制源代码

1.2 避障源代码

1.3 路径规划源代码

1.4 目标检测源代码

1.5 目标抓取源代码

2. 可执行程序

2.1 机器人基本运行程序

2.2 机器人路径规划导航程序

2.3 机器人目标抓取程序

3. 设备

课程组提供的硬件设备

### 文档

1. 项目开发计划

对项目生命周期内的工作范围、工作任务的拆解、项目组织结构以及人员分配、开发进度、项目风险评估、项目成本与预算、人力资源计划等内容以书面形式做出安排。项目开发计划为实际的项目开发过程做出全局性的规划指导，并成为项目参与人员的行动共识，作为团队对项目进行开发和检查的依据。

1. 需求分析文档

详细的对简易机器人项目做出全面详细的用户需求分析，明确项目的功能与性能。与用户沟通达成对项目成果一致的理解，并为后续的设计计划打好基础，使得软件设计更好地符合用户需求。

1. 软件设计文档

根据《需求分析文档》中描述的功能、运行和性能需求制定详细的项目设计。提供统一的参照标准，包括机器人内外部接口、系统架构、编程模型以及其它主要问题的解决方案。后续的实际开发将遵循设计文档的相关标准进行。

1. 测试文档

分为单元测试文档和集成测试文档，为项目制定全面有效的测试计划，供软件测试人员在具体测试时参考。单元测试项目的软件构件，进行局部化的验证。集成测试项目多个构件集合，进行功能构件之间交互性的测试。文档具体内容包括但不限于确定项目测试构件、列出测试需求、推荐并说明测试策略、确定所需资源、明确测试工作量及可交付元素等内容。

## 运行与开发环境

### 运行环境

1. 硬件环境

嵌入式开发板、各类传感器、运动装置、机械臂装置及启智ROS基本组件。

1. 软件环境

Ubuntu16.04和ROS kinetic。

### 开发环境

1. 硬件环境

嵌入式开发板、各类传感器、运动装置、机械臂装置及启智ROS基本组件。

1. 软件环境

Ubuntu16.04、RoboWare Studio及相关基本运行库等。

## 项目期限

项目起始时间：2020.03.09

项目结束时间：2020.06.17

# 风险管理

表3-1 风险评估与管理策略表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 风险识别 | | | 风险管理策略 | |
| 风险类型 | **风险举例** | **发生可能性** | **应对措施** | **预防措施** |
| 需求 | 需求不明确 | 中等 | 与客户进行全面细致的交流沟通，明确需求，做出正确描述 | 事先进行需求评审 |
| 后续需求调整 | 较高 | 提交CCB讨论、决定 | 建立需求范围变更程序 |
| 缺乏有效的需求变化管理过程 | 中等 | 对需求变化进行评审 | 建立需求变更程序 |
| 估算 | 总体开发时间过长 | 较高 | 根据实际情况重新安排各阶段时间 | 规划科学可行的开发计划 |
| 各任务持续时间不均 | 较高 | 调整人员分配和任务分配 | 对任务的划分尽量均匀 |
| 人员分配工作量不均 | 中等 | 对工作量进行适当调整 | 对工作量划分尽量均匀 |
| 技术 | 开发平台不稳定 | 较高 | 及时稳定调整 | 开发前尽量了解开发平台特性和相关应对策略 |
| 可选构件集不完整 | 较低 | 调整设计适应现有构件 | 开发前了解可选构件 |
| 数据库性能不高 | 中等 | 调整数据库架构设计等 | 加强数据库架构设计 |
| 人员 | 人员出现变动 | 较低 | 重新分配任务，调整人员分配 | 增强团队凝聚力 |
| 技术水平不够 | 较高 | 及时进行培训或调整人员分配 | 加强个人技术水平 |
| 沟通机制低效 | 中等 | 改变沟通机制并解决因沟通不完备造成的程序问题 | 及早摸索适合团队的高效沟通机制 |
| 测试 | 测试期间出现重大问题 | 较低 | 及时修改程序 | 单元测试与集成测试 |
| 测试计划不可行 | 较低 | 修改测试计划 | 事先评审测试计划 |

# 过程模型

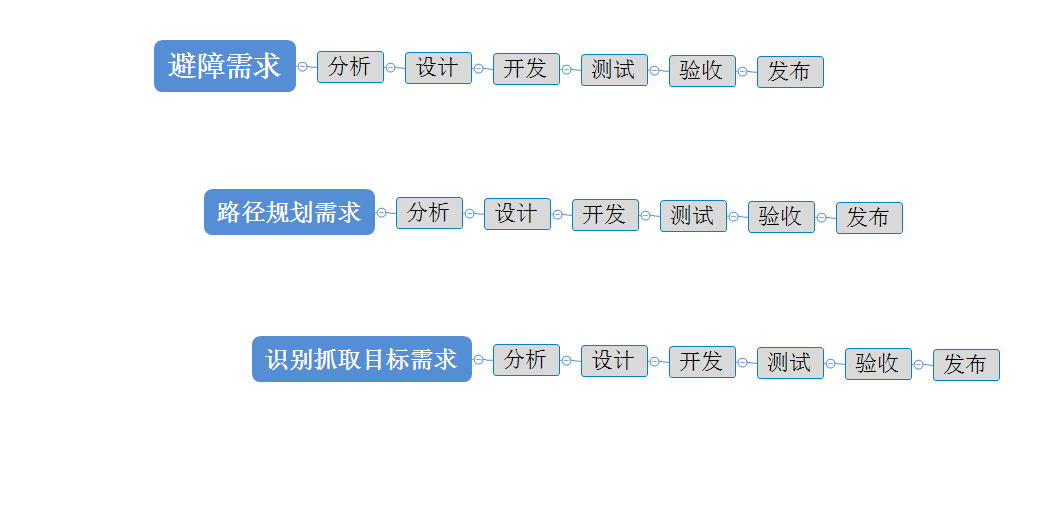
本项目开发采用迭代-增量生存周期模型。

图4-1 迭代-增量模型

项目预计划分为三个需求：避障需求、路径规划需求及识别抓取目标需求。即成为三个主要阶段，使用随时间进展交错的线性任务排列进行开发，每个序列产生一个可发布的增量，每个增量产生一个可操作的产品。在需求阶段就给出相对完整的架构设计方案，灵活安排人员分配。在第一阶段的核心功能交付后可以得到可运行的软件雏形，及时获得用户反馈并改进调整后续开发计划。

# 资源计划

1. 硬件资源需求

搭载摄像机、各类传感器、运动装置、机械臂装置等硬件的启智ROS机器人。

1. 软件资源需求

Ubuntu系统、启智ROS机器人基于Ubuntu16.04的kinetic版本ROS系统、RoboWare Studio及相关运行库。

# 进度计划

## 里程碑事件

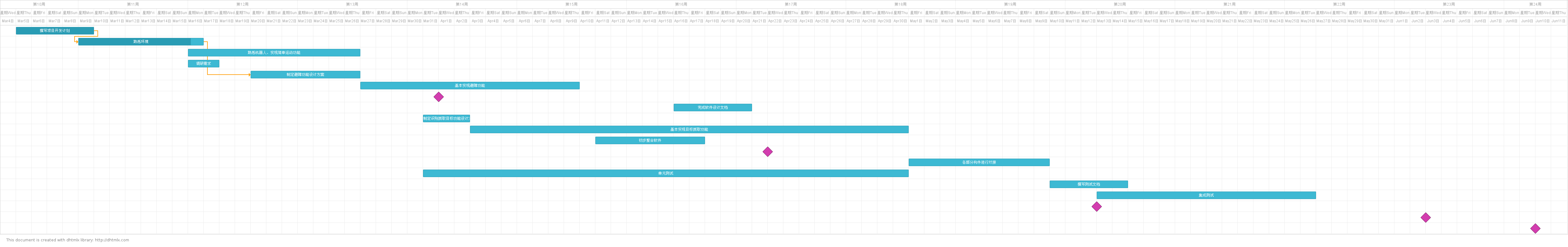
表6.1-1 里程碑事件

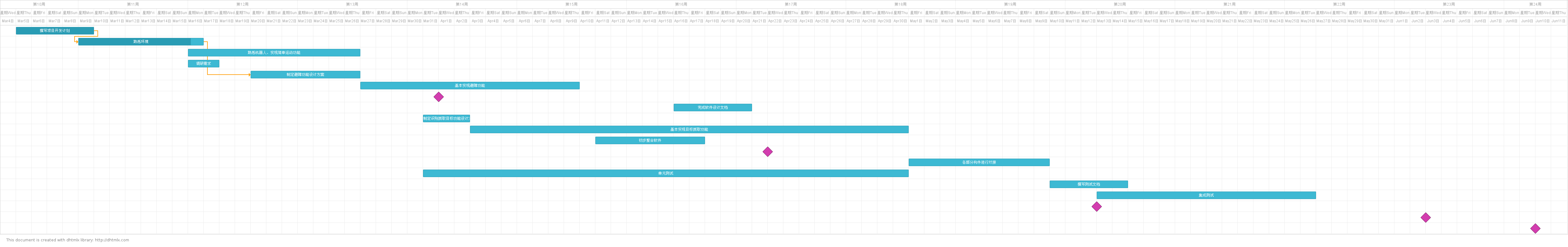
|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 时间节点 | 里程碑事件 | 项目状态 |
| 1 | 2020.03.09 | 提交项目开发计划 | 项目开发计划已完成 |
| 2 | 2020.03.15 | 前期准备 | 配置开发环境，调研分析产品需求，进行初步分析 |
| 3 | 2020.03.29 | 完成需求分析 | 各成员了解各自分工及相关知识 |
| 4 | 2020.03.30 | 提交需求分析文档 | 需求分析文档已完成，准备需求分析评审presentation |
| 5 | 2020.04.01 | 需求分析评审 | 小组/个人presentation |
| 6 | 2020.04.10 | 初步完成粗略软件设计 | 细化产品需求和软件设计 |
| 7 | 2020.04.20 | 提交软件设计文档 | 软件设计文档已完成，准备软件设计评审presentation |
| 8 | 2020.04.22 | 软件设计文档评审 | 小组/个人presentation |
| 9 | 2020.05.01 | 各部分构件开发基本完成 | 各部分开发基本完成 |
| 10 | 2020.05.02 | 进行单元测试 | 单元测试并进行调整 |
| 11 | 2020.05.05 | 各部分构件进行对接 | 构建系统 |
| 12 | 2020.05.11 | 系统稳定性和正确性基本合格 | 准备代码评审内容 |
| 13 | 2020.05.13 | 代码评审 | 系统基本合格 |
| 14 | 2020.05.14 | 编写测试文档，完善各部分构件 | 集成测试并调整 |
| 15 | 2020.06.01 | 集成测试基本完成 | 系统正确性进一步提高，准备测试文档第一次评审 |
| 16 | 2020.06.03 | 测试文档第一次评审 | 准备测试文档第二次评审 |
| 17 | 2020.06.10 | 测试文档第二次评审 | 进一步完善系统和文档 |
| 18 | 2020.06.17 | 课程总结 | 小组/个人总结 |

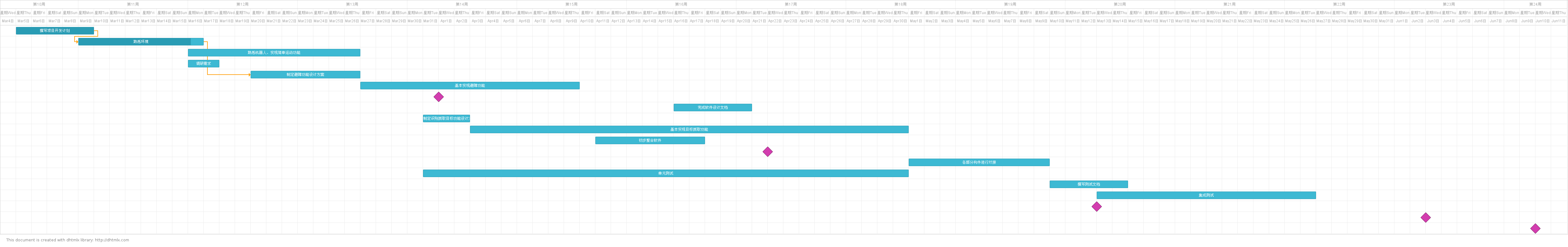
## 里程碑事件人员分配

表6.2-1 里程碑事件人员分配

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 时间节点 | 里程碑事件 | 人员分配 |
| 1 | 2020.03.09 | 项目开发计划 | 全体 |
| 2 | 2020.03.15 | 配置环境  熟悉ROS  阅读ROS开发手册  明确分工内容 | 全体 |
| 3 | 2020.03.17 | 调研使用需求以及相关数据参数  完成需求分析文档 | 孙一丹、潘林煜、施雨萌 |
| 4 | 2020.03.25 | 熟悉机器人基本功能  搭构框架，基本实现简单运动功能 | 郑奕、潘林煜 |
| 5 | 2020.03.27 | 制定避障功能设计方案 | 施雨萌、郭若熙 |
| 6 | 2020.03.31 | 准备评审资料 | 潘林煜、施雨萌 |
| 7 | 2020.04.05 | 基本实现避障功能  制定识别抓取目标功能设计方案 | 施雨萌、郭若熙  孙一丹、郑奕 |
| 8 | 2020.04.10 | 初步完成粗略软件设计整合 | 全体 |
| 9 | 2020.04.20 | 完成软件设计文档  准备软件设计评审材料 | 潘林煜、施雨萌 |
| 10 | 2020.04.30 | 基本实现识别抓取目标功能 | 孙一丹、郑奕 |
| 11 | 2020.05.01 | 各部分构件开发基本完成  开始单元测试 | 全体 |
| 12 | 2020.05.05 | 各部分构件进行对接 | 全体 |
| 13 | 2020.05.11 | 准备代码评审材料 | 潘林煜、施雨萌 |
| 14 | 2020.05.14 | 编写测试文档 | 郭若熙、潘林煜、施雨萌 |
| 15 | 2020.06.01 | 集成测试基本完成 | 郭若曦 |
| 16 | 2020.06.03 | 准备测试文档评审 | 潘林煜、施雨萌 |
| 17 | 2020.06.16 | 准备测试文档第二次评审 | 全体 |
| 18 | 2020.06.17 | 课程总结 | 全体 |







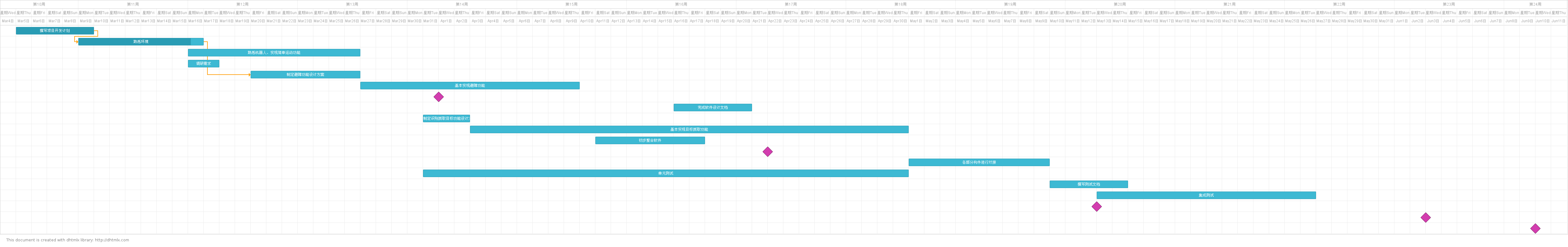


图6.2-2 甘特图

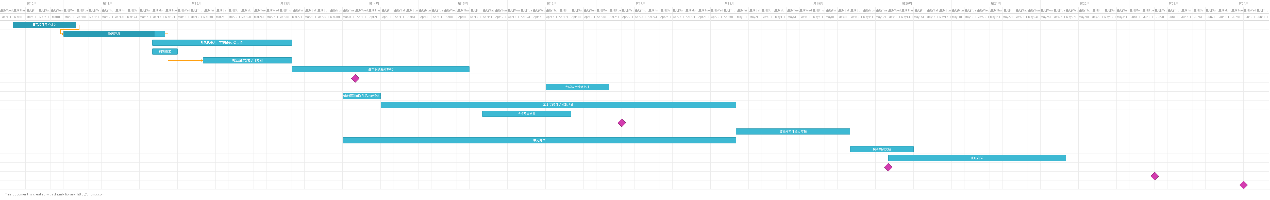


图6.2-3 甘特图缩略图