**简易机器人**

**嵌入式软件工程课程项目总结报告**

**v1.0.0**

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 |
| 16061052 | 周环宇 | 第5，6，7，8章相关部分以及第10章 |
| 16061074 | 朱洪东 | 第5，6，7，8章相关部分以及第9章 |
|  |  |  |
|  |  |  |
|  |  |  |

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 项目名称

搞事班的故事：简易机器人

# 项目概述

【简述项目的目标、功能和非功能需求，最终完成的功能，达到的效果等】

# 需求分析阶段总结

【针对需求分析活动和需求评审所发现的问题进行总结，概述需求分析采取的策略和面临的困难，以及后续阶段是否对需求提出了修改要求】

# 设计阶段总结

【针对设计活动和设计评审所发现的问题进行总结，概述系统架构，以及设计到需求的追踪关系。分析设计阶段所发现的问题】

在总体的系统架构上，本产品主要实现两个功能，即扫地和取物。为了实现这两个功能，需要底层的数个基础功能模块的技术支撑，这些模块包括运动模块，建图模块，路径规划模块，目标识别模块，抓取模块，总控模块，传感器数据获取模块等。在这些模块里，运动、建图、路径规划、总控以及传感器数据获取模块是两个功能公用的，抓取和目标识别是取物功能独有的。在实现了这些模块的基础上，我们将其进行组合以实现机器人的两主要功能。除此之外还有数据库的设计，数据库的主要功能为存储机器人的运行日志信息。日志具体分为五类，分类是清扫日志、取物日志、开关机日志、错误日志以及其他日志，每种日志都有一张自己单独的表构成了整个机器人的数据库。在需求可追踪性方面，由于机器人的主要功能由模块组合实现，所以可追踪性主要体现在各个模块上：基础运动由MoveController类实现，路径规划由GetEnvironment, GenMap, Navigation, MapController类共同实现，物体识别由TargetDetect, SceneGetter, TargetDetectControl共同实现，物体抓取由GrabItem, MoveBack, GrabControl, MechanicalArm共同实现该功能，日志处理由exception类实现。从现在看来，我们在设计阶段发现的问题如下。第一是在做需求用例图的时候忽略了传感器的存在，导致用例图不够完整，不能很好的说明程序的功能性；第二个就是部分模块设计的冗余以及有些模块之间存在进程冲突的问题，这些问题都在后续的代码中得到了相应的解决。

# 迭代1阶段总结

【从任务分配、代码提交、测试和问题管理四个方面总结该阶段的团队开发进展情况，以及评审时所发现的问题和相应的改进措施】

## 5.1 任务分配

**周环宇：**扫地模块开发，使用dfs生成扫地的路径，主要是引用了move\_base包。机器人行走过程中除了考虑全局的静态边界以外，外主动避开局部的动态障碍物。标记过的点不会再次扫描，dfs生成的路径自带回溯，最终机器人会回到起始点。

**刘博文：**扫地功能的测试部分，包括对扫地路径等一些输入参数以及扫地效果等方面的测试

## 5.2 代码提交

**周环宇：**

d6624af069d2762dc6b229b72d4aae0697cfb528（加入扫地模块（版本1）的源代码）

## 5.3 测试

**扫地模块初步测试：**首先建好地图，然后在封闭环境下运行扫地模块，机器人能既可能的遍历一遍可达点，然后返回原处。

## 5.4 问题管理

**扫地模块目前问题**：机器人初始位置有一定的限制，不能里地图原点太远；遇到障碍物或边界的时候机器人会原地徘徊一阵子才能抉择出下一个方向；机器人行动过程中自动定位存在一定的偏差，最终回到的点和出发点有一定的累计误差。

## 5.5 评审时所发现的问题和相应的改进措施

**缺少完整的看板开发管理：**团队成员都新建worktile账号并加入到对应的项目中，便于团队开发管理。

# 迭代2阶段总结

## 6.1 任务分配

**周环宇：**扫地模块完善，目前的dfs生成的路径是“绕圈”式的，即由外圈走向内圈。 现在新增了一种zig-zig式的，走“之”字，还没有进行测试。考虑加一种强清洁模式，机器人边走边反复蹭某个地方两三次然后在继续走。

**刘博文：**解决机器人日志提取、存储以及前端日志页面的显示功能

## 6.2 代码提交

**周环宇：**

09703939eb54f66dd987221d27d02cf1949428cf（两种遍历模式：绕圈型与之字型；清扫强度等级；语音控制停止与继续）

33a17a944c52347d6b680e1f531230c445074b2c（扫地模块：输出改为ROS\_INFO，test\_dfs调试出错误）

3196a827fa00a030e187dd73702dcf83637ff13f（减少遇到地图边界时的迂回时间，新增订阅/rosout的回调函数，收到log中err级别的信息时即可判断为plan failed，提前枚举下个方向。）

**刘博文：**

ff2c8d41c9e4aab75ba99978fce6ab28fbd0f37b

8d256b71ec8e0edda560e018c2caaec87ef207c4

## 6.3 测试

**扫地模块路径生成测试代码：**

<https://github.com/sebuaa2019/Team109/blob/master/RobotSrc/clean_module/src/test_dfs.cpp>

**扫地模块语音测试代码：**

<https://github.com/sebuaa2019/Team109/blob/master/RobotSrc/clean_module/src/test_voice.cpp>

**日志模块功能测试：**

测试主要针对以下几个部分，一是日志的读取功能，即从rosout.log文件中提取出日志的信息，这一部分是一个简单的文件读取函数，测试则是用不同内容的文件查看读取的完整性；二是日志存储，这一部分主要是存储在函数内，所以主要测试list的功能；三是网页前端的显示问题，使用了分页器，所以主要测试分页器的功能，测试集中在views.py文件。

## 6.4 问题管理

<https://github.com/sebuaa2019/Team109/issues/1> (worktile的看板与github的commit关联起来)

<https://github.com/sebuaa2019/Team109/issues/2> (关于单元测试)

**日志的一些问题：**存储在本地的log不具有前端部署的实时性，下一步应将挪入数据库。日志信息单一，下一步应提取出更多的信息。日志时间戳不够直观，下一步应将其转换为世界时间

## 6.5 评审时所发现的问题和相应的改进措施

**看板与github不关联，git中的issue没有充分利用：**团队成员分别完善各自下的worktile任务内容，具体到git上的commit。所有的讨论转移到git中的issue中进行。

**只有扫地模块有单元测试：**要求其他成员参照扫地模块尽快补全相关测试代码。

# 迭代3阶段总结

## 7.1 任务分配

**周环宇：**扫地模块再完善以及前后端结合相关，launch文件合一，每个功能的调用只会run一个node，扫地模块根据需求，需要统计面积、距离、时间。

**刘博文：**完善机器人的日志功能

## 7.2 代码提交

**周环宇：**

8e95d882eb2919adb4fec2ed2060b8c4a60dfb63（扫地模块新增统计面积、距离、时间，写入文件~/catkin\_ws/clean.out中，格式为%lf %lf %d。提交总体的launch第一个版本）

889d0b0905e86e6fb2e1ff0dd700f4b521fd16c0（调大了扫地初始化的延时等待。新增core\_109，核心依赖launch core\_109.launch）

ff4b568dce6131b065e44ac0f6e733d50673a2bd（消除rosrun指令，各个模块只留launch）

**刘博文：**

778e8d85e6daf75e27aab7b501718769c324693c

4175d7f380050fc55ec6460435ea961c81dca3d6

7cc2c8716809a7249428344941484e783dce2416

bb544b9b449859f3d62beb98672fbcc0d90824ac

## 7.3 测试

**扫地模块综合测试：**测试扫地的不同扫地模式，不同扫地强度以及语音控制停止与继续，对于障碍物的规避。

**日志模块测试：**主要测试数据库存储功能，由于Django的数据库由models.py管理，所以测试主要针对这个文件，通过读写数据库来测试功能是否完善

## 7.4 问题管理

<https://github.com/sebuaa2019/Team109/issues/3> (关于launch文件)

<https://github.com/sebuaa2019/Team109/issues/4> (launch指令)

<https://github.com/sebuaa2019/Team109/issues/5> (关于扫地的功能)

<https://github.com/sebuaa2019/Team109/issues/6> (关于实现对机器人控制的讨论)

<https://github.com/sebuaa2019/Team109/issues/7> (标签征集)

<https://github.com/sebuaa2019/Team109/issues/8> (关于界面上的多余内容)

<https://github.com/sebuaa2019/Team109/issues/9> (冲突)

<https://github.com/sebuaa2019/Team109/issues/10> (About界面的自我介绍)

<https://github.com/sebuaa2019/Team109/issues/11> (Log与实时信息显示)

<https://github.com/sebuaa2019/Team109/issues/12> (大launch 来了DAZE✨！！)

<https://github.com/sebuaa2019/Team109/issues/13> (数据库报错了，报错信息如下)

<https://github.com/sebuaa2019/Team109/issues/14> (合并之后的问题)

<https://github.com/sebuaa2019/Team109/issues/15> (第三次迭代后老师提出的问题)

**日志的一些问题：**前端美观度欠佳，可以通过使用模板解决

## 7.5 评审时所发现的问题和相应的改进措施

**测试的改进**：需要统计覆盖率，用例的设计考虑边界情况等等，可以有课件ppt参考，相关组员调研测试工具并完善测试用例的设计

**文档的完善**：文档与实际的区别，开发过程中对需求与设计的改动在文档中跟进改动

# 测试总结

【从测试角度，从需求覆盖和设计单位（如类、函数或接口等）两个层面来梳理采取的测试策略，并概述设计的测试用例，阐述测试效果，以及测试所发现的问题】

## 8.1 测试策略

从需求覆盖角度，为了覆盖机器人扫地功能这一需求，设计了三个测试用例，001（扫地单元路径生成），002（扫地单元语音识别），003（扫地单元集成）。其中001是对于扫地路径生成函数（dfs）的单元测试，002是对于扫地语音回调函数（voiceCB）的单元测试。

从需求覆盖角度，为了覆盖机器人运行日志显示这一需求，设计了测试用例009，用于测试整体日志模块。

## 8.2 测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 需求 | 名称 | 编号 | 描述 |
| 扫地功能 | 扫地单元路径生成 | 001 | 扫地时进行路径规划 |
| 扫地单元语音识别 | 002 | 通过语音识别控制机器人停止或重新开始 |
| 扫地单元集成 | 003 | 扫地功能完整流程 |
| 建图功能 | Socket单元 | 004 | 通过socket进行运动控制交互 |
| 运动控制与建图 | 005 | 进行运动控制的同时建图并保存建图的结果 |
| 用户交互 | 扫地功能界面交互 | 006 | 通过网页输入信息调用扫地功能 |
| 取物功能界面交互 | 007 | 通过网页输入信息调用扫地功能 |
| 界面显示 | 008 | 在手机和电脑显示各模块界面 |
| 可维护性 | 日志 | 009 | 读取log文件，显示日志信息 |
| 取物功能 | 取物单元标签 | 010 | 识别取物用的标签 |
| 取物单元抓取 | 011 | 对识别到的空间坐标进行抓取 |
| 取物单元集成 | 012 | 扫地功能完整流程 |

## 8.3 测试效果

001：符合预期

002：符合预期

003：与预期存在偏差

009：符合预期

## 8.4 测试所发现的问题

在扫地综合测试时，机器人会基本按照对应的mode和level行走，但路径存在偏差。机器人整体运行过程中基本会躲避障碍物（包括静态的以及动态的），但有些障碍物无法躲避。

# 团队协同总结

【从团队协同角度，概述团队成员的分工模式，并梳理各个成员实际承担和完成的任务。并对本项目团队的协同效果进行评价】

团队

文档最初计划

开发计划 朱洪东

需求文档 周环宇

设计文档 金阳 周环宇

测试文档 王闯 刘博文

后来发现这样分工会导致每个人在一个阶段很忙，任务量很大，不能较好地完成任务，而在其它时间段有没有事情可做。所以写了第一次的开发计划后，就在修改时将开发计划重新分工并写了一遍。并且之后的文档也都是这样来完成。分工如下方所示。

开发计划

分工

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 16061074 | 朱洪东 | 修改风险部分。 |
| 16061080 | 王闯 | 修改范围部分。 |
| 16061088 | 金阳 | 修改里程碑部分。 |
| 16061038 | 刘博文 | 参与讨论。 |
| 16061052 | 周环宇 | 修改模型部分。 |

当时开发计划的修改和需求文档同时进行，开发文档没有多余可修改的部分，所以负责在需求文档负责两章。

这样按章分工后，首先是显而易见地，字数很容易就达到足够的量：各人想法不同，而且精力要比只有一个人完成一个文档时更多。

需求文档

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 16061074 | 朱洪东 | 范围 |
| 16061080 | 王闯 | 数据需求 |
| 16061088 | 金阳 | 业务需求 |
| 16061038 | 刘博文 | 非功能需求 运行与开发环境 |
| 16061052 | 周环宇 | 功能需求 |

这里发现功能需求要比其它部分工作量大。

设计文档

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 16061074 | 朱洪东 | 需求概述、可追踪性说明 |
| 16061080 | 王闯 | 数据库设计 |
| 16061088 | 金阳 | 详细设计 |
| 16061038 | 刘博文 | 软硬件接口 |
| 16061052 | 周环宇 | 体系结构设计 |

吸取设计文档的经验，这次将可能比较困难的详细设计和体系结构设计进行了小组讨论之后才开始写。这样任务完成得更好。

迭代阶段1

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 工作内容 |
| 16061074 | 朱洪东 | 抓取部分:修改launch |
| 16061080 | 王闯 | 抓取部分:识别 |
| 16061088 | 金阳 | 灵活调动🡪手柄移动 |
| 16061038 | 刘博文 | 扫地部分 |
| 16061052 | 周环宇 | 扫地部分 |

这个阶段由于组员对git不熟练，所以都是在最后进行commit，所以commit数不多。

这个阶段扫地和手柄移动实现了，但抓取还不能使用，做到了将标签识别出来。

迭代阶段2

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 工作内容 |
| 16061074 | 朱洪东 | 扫地和抓取界面 |
| 16061080 | 王闯 | 抓取 |
| 16061088 | 金阳 | 网页操控及操控界面 |
| 16061038 | 刘博文 | 日志部分 |
| 16061052 | 周环宇 | 扫地部分 |

这个部分界面设计和实现都已完成，但还不能结合使用。后来发现，这时候在分工上出现了一点问题：操控部分由于和界面时一人负责，所以结合较好，而扫地和抓取与网页的结合其实不好，这个时候应该更多地交流。

迭代阶段3

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 工作内容 |
| 16061074 | 朱洪东 | 界面完善和交互 |
| 16061080 | 王闯 | 抓取 |
| 16061088 | 金阳 | 界面完善和交互，用户注册登录 |
| 16061038 | 刘博文 | 日志部分 |
| 16061052 | 周环宇 | 扫地新增模式 |

由于最后的集成，这个部分时全组不断交流完成的。同时，这个阶段较多地使用了issue，也促进了交流。

测试文档

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 16061052 | 周环宇 | 第3、4、5、6章中对扫地模块相关的测试 |
| 16061074 | 朱洪东 | 第1、2章，第3、4、5、6章中对界面相关的测试 |
| 16061080 | 王闯 | 第3、4、5、6章中对取物模块相关的测试 |
| 16061088 | 金阳 | 第3、4、5、6章中对移动与建图相关的测试 |
| 16061074 | 刘博文 | 第3、4、5、6章中对日志提取的测试 |

测试阶段由于各个人的单元测试要自己完成，所以文档也时每个人写自己的部分。

各阶段PPT类似，基本与文档分工对应。

# 感受与建议

【阐述团队成员对课程及课程项目的感受，并为课程改进优化提供三个具体的建议】

## 10. 1 团队成员的感受

**朱洪东：**身为组长，第一次组织团队共同开发一个工程项目，从技术角度学到了ros这个机器人系统工具，从管理角度意识到了看板上任务流程管理，issue问题管理等东西在整个团队协作过程中的作用，锻炼了团队协作能力，学会了如何给成员分工，如何组织成员协作，以及监督成员的工作。

**金阳：**这门课教会了我一个完整的软件工程开发过程，从需求分析，到设计，到开发，到测试，环环相扣，每个部分都需要投入大量精力，而且少不了团队合作，团队之间的沟通交流。从编程的角度来说也是锻炼了我个人的网页开发能力，尤其对Django更加熟悉，才了不少坑，也学习了许多机器人相关的知识，享受到了真正开发一个产品的快乐。

**王闯：**我主要是负责机器人后端模块的物体识别与物体抓取部分，虽然这部分有机器人自己的代码，但我们的需求不同，需要识别特定的标签，而且机器人的机械臂的偏移也很容易出错，需要多次调整参数。通过这门课程我学到到了功能开发的不易，以及测试的重要性，全面的测试才能保证功能的完美展现。

**刘博文：**软工是一门强调团队合作的课程，如何在团队中最大化每个人的作用是很重要的，这里不开精细化的分工，每个人根据自身能力承担应有的责任。在软工开发的过程中，我学习到了许多技术，尤其我负责的工作中的django网页开发以及数据库的调用，日志信息的筛选与整理，以及测试中的用例设计对于一个工程的重要性。

**周环宇：**软工告诉了我团队协作的重要性，工程的开发一个人蛮干是不可能的，自己的部分也不能随心所欲的按照自己的想法来写，要考虑到自己的上游与下游。我是主要负责机器人后端的开发，扫地功能的实现，不仅仅要考虑实现的展示效果，还要给前端调用后台命令提供更好的方便性。

## 10.2 对于课程的建议

**前期的设计与后期的开发：**对于软件工程开发而言，需求分析与系统设计确实是很重要的，但就整个课程的进度上来看，前大半部分都是在进行文档方面的答辩，有的文档留出了两周时间来写，而真正的开发只用了三周迭代的方式来完成，当然这样与器材到的比较晚有关。具体来说可以讲前期设计的时间压缩一点，可以将某几个文档合成一个，比如需求文档和设计文档合并起来，这样也可以加深两个文档内容上的关联紧密性。从而预留给更多的时间给每组以自己的方式进行各自的开发计划。

**引入周报与周会：**从团队开发的角度而言，有两个重点要把握的地方，一个是团队内的交流沟通，还一个是进度的监督把握。这次课程中有的组就已经做好每周开会并且有会议记录的习惯，可以推广开来。让每个组都留下每周的会议记录，这次会议中主题是什么，解决了什么问题，制定了什么计划，团队成员的发言等等。而每周的周报就是要汇报每个人这周完成了什么，下周的计划是什么，这样也是可以推动个人的执行力以及方便后续相关数据的统计。

**关于文档答辩时间过长：**每次文档答辩的时间都会拖的很长，导致后面的组无法在课堂时间内讲完，而对于这一部分的内容其实大部分组只是在单纯的搬挪文档上的字。这一部分也可以改成类似申优答辩的形式，对自己的设计觉得有新意的组可以上台分享一下小组的思路，如果单纯只是文字描述就可以解决的话就通过文档评定即可，不需要专门上去复述文档内容。多出来的时间可以用来课堂知识的讲授。