嵌入式软件工程课程项目总结报告

## 项目名称

机器人操纵的辅助APP

## 项目概述

（1）目标：本项目基于启智机器人开发平台，实现了一个机器人操纵的辅助APP，实现了对机器人的远程操控和路径规划功能。

（2）功能需求

功能一：用户通过手机APP可操控机器人运动，期间机器人完成对环境的建图，同时可以切换信号源，查看机器人摄像头数据和建图情况

功能二：基于已经建好的地图，用户可以指定目标点，机器人完成多点导航

（3）非功能需求

性能需求：优化路径规划算法，提高机器人的响应速度

可靠性需求：针对一些异常状况可以做出合理的反应，比如突然出现障碍物机器人将急停

（4）完成的效果

已实现手机端的操纵，可以实现前进后退左右旋转。

可以从手机端查看摄像头数据。

可以从手机端查看机器人建图数据。

可以基于建好的地图实现多点导航。

缺点：对于图像传输延迟较高，大概5s左右。在导航行进过程中，速度较慢。

## 需求分析阶段总结

项目采用基于场景的模型对需求进行建模。

第一阶段我们想自主实现机器人的基本避障运动，实现方法为设定机器人运动的方向和距离，通过前进、躲避障碍物、恢复起始运动方向这一系列动作到达设定的目的地。为完成这一需求，设计了四个大的用例：机器人行使/起停，机器人规避障碍，机器人确定相对位置，重新配置传感器及相关系统特性。此阶段的问题是对机器人硬件和功能了解不足，用例的设计脱离机器人所能够提供的基本功能。

第二阶段在开始设计之前，我们对需求进行了更改，想要实现三个大的模块：机器人避障运动，主动控制运动，路径规划运动。设计了五个大的用例：基本避障，主动控制，路径规划，异常状况，重新配置传感器与相关系统特性。此时对机器人有了一定的了解，知道了机器人运动控制和获取雷达数据的方法，并计划利用已有的一些包实现我们的功能。

第三阶段中段总结反思时，我们确定了将对机器人的控制集成到APP上，实现两个功能：机器人的操控移动和机器人的多点导航，同时可以从手机端获取建图和摄像头数据，将本次项目定位于面向用户的机器人操纵辅助产品，利用已有的一些包改造之后实现自己的功能。

## 设计阶段总结

设计活动和设计评审中存在的问题：只是了解了大概的实现方向，设计比较笼统，与能够直接开始开发还有很大的距离。设计所依赖的需求还没有最终确定，之后改变需求时对设计又进行了修改。

系统架构：技术实现方面基于ROS的通讯框架，通过订阅若干Topic实现机器人控制、定位、雷达信息获取功能。硬件方面基于ROS操作系统，通过接口与处理器和其他外设连接。

需求追踪：机器人运动控制通过向/cmd\_vel发布控制信息实现。用户输入控制通过APP实现。路径规划通过调用包实现。

## 迭代1阶段总结

任务分配：由刘华兵负责机器人远程控制，陈强负责机器人图像传输。

代码提交：分别由各自在第一次迭代总计报告前提交（commit ）。

代码测试：由个人独自对代码先进行测试，基本功能实现后提交到github上，小组统一去实验室检测代码在机器人上是否能正确执行、完成目标功能。

问题管理：实现过程中发现一个问题，并提交到issue上，在代码实现过程中遇到的问题由各部分负责人解决，并没有体现出团队问题管理，没有将问题提交到github上进行问题管理，在测试中遇到他人代码出现问题，由组内成员私下交流解决软件功能问题。

开发进展情况：团队整体开发进度满足第一次迭代预期功能要求。

评审中问题：团队开发细节不到位，软件开发过程不清晰透彻，团队中各个组员的贡献不明确，项目开发过程中的问题处理没有体现出来，整体项目开发进度不透明可视化。

改进措施：避免一次性最终提交代码，代码提交过程的可实化。

## 迭代2阶段总结

任务分配：由刘华兵负责机器远程建图和导航实现，陈强负责地图实时建图传输。

代码提交：分别由各自在第二次迭代总计报告前提交（commit ）。

代码测试：由个人独自对代码先进行测试，基本功能实现后提交到github上，小组统一去实验室检测代码在机器人上是否能正确执行、完成目标功能。

问题管理：在各自代码实现过程中对代码进行测试，并没有将问题提交到github上，在测试中遇到他人代码出现问题，由组内成员私下交流解决软件功能问题。

开发进展情况：团队整体开发进度满足第二次迭代预期功能要求。

评审中问题：软件开发过程不透明清晰，缺少看板对项目的管理情况展示，没有对问题管理进行关注。

改进措施：实现对项目的看板管理，细化各组员实现的功能任务。

## 迭代3阶段总结

任务分配：由胡峰完成测试代码。

代码提交：分别由各自在第三次迭代总计报告前提交（commit ）。

代码测试：由个人独自对代码先进行测试，基本功能实现后提交到github上，小组统一去实验室检测代码在机器人上是否能正确执行、完成目标功能。

问题管理：在各自代码实现过程中对代码进行测试，并没有将问题提交到github上，在测试中遇到他人代码出现问题，由组内成员私下交流解决软件功能问题。

开发进展情况：只完成了部分的测试代码。

评审中问题：软件开发过程中各组员之间的问题管理和交流并没有体现在github上，没有对项目进行细致的动态管理。

改进措施：要求各组员积极在网站提交在实现过程中的问题，完成各自分配的任务。

## 测试总结

测试策略：代码测试的角度由两方面完成，一方面是作为测试人员主要测试代码是否可执行，是否实现了具体的功能。另一方面是用户角度，测试代码实现对于用户的体验，用户对于软件功能实现的评价。代码测试主要为对代码的功能需求进行测试，测试代码是否能完成预期实现的功能，测试覆盖了所有的功能需求，对于非功能需求，团队主要对于实时性进行了测试，检测远程操控和传输的延迟。测试设计单位为函数，对于具有代表性的功能函数进行测试，检测功能函数是否正确实现其具体功能。

测试用例：测试用例包括客户端与服务端的连接测试用例、图像传输测试用例、机器人远程控制传输测试用例、地图信息传输测试用例、机器人建图与传输测试用例、机器人远程导航测试用例等，以上测试用例主要从项目的功能需求进行测试。所有的测试的输出与预期一致，但测试中还有问题，测试发现图像传输的延迟，机器人摄像机的实时视频在移动端具有延迟，机器人导航过程中运动缓慢。

## 团队协同总结

团队分工模式：根据项目实现的要求，各组员分别负责项目的不同功能，各组员完成不同的项目工作。

项目成员实际承担任务：刘华兵负责机器人在移动端的远程控制、远程机器人导航和导航信息转换，陈强负责机器视频图像、机器人建图实时传输的功能，胡峰负责完成测试代码，各组员完成独立的功能模块。

团队协同效果：各组员分别完成不同模块，功能独立，因此不会受到彼此的影响，利于各自独立完成任务。

## 感受与建议

刘华兵：经过软件工程一学期的理论学习与代码实践，我理解了在工程实践中软件的开发流程，通过Github的使用，理解了工程开发中的团队合作，同时，在ROS的基础上，通过编写服务器功能节点，完成了Android手机对机器人的远程操控，代码能力进一步提高。

陈强：在这次课程学习过程中经历了许多，在项目开发过程中遇到许多突发情况，面对项目进度因为各种原因的延期，需要不断的调整每个人的任务和项目的工作目标。作为组长，我觉得软件开发是一个团队合作项目，项目的管理更优于代码的实现，软件开发的重点不在于写出完美的代码，更优质的团队管理和合作是本次课程对于我最大的影响。

胡峰：课程项目团队开发与个人作业开发有很大不同，简单地将整个项目割裂为一个个小部分由个人完成，由于缺乏交流个人理解上有偏差，项目容易出现各部分不一致，影响项目的进度和质量。真正的团队开发应该在团队成员有统一的认识的前提下，合理分配任务，及时的交流沟通十分重要。

建议：

1 在团队建立的时候，应当考虑实际的非面对面的开发场景，大多数情况下同学们开发过程都是面对面，因此忽视了对于过程的记录。相互认识的人不利于项目的推进，容易出现抱大腿、划水现象，对于团队组成应当从开始慎重决定。

2 我希望将课程的考核标准从课程开始提出，利于小组重视项目开发的过程，有着一定的审核标准有利于同学早期了解课程对软件开发的要求，对软件开发过程有更深的理解，减少团队只重视功能的实现疏于小组管理等问题

3 对一些工程化方法的讲解可以多与实践相结合，通过实践学习，减少单纯文字讲解所占比例。