****

**本科生毕业论文（设计）**



**中文题目： 基于可穿戴式设备的人-机器人协作研究**

**英文题目：Research on the man-robot cooperation based on the wearable devices**

**姓名学号： 潘世民 3150102585**

**指导教师： 杨赓（浙江大学）王菲（中国美术学院）**

**张英（中国美术学院）**

**年级与专业： 2015级 机械电子工程**

**所在学院： 机械工程学院**

2019年 6 月 10 日

**浙江大学本科生毕业论文（设计）诚信承诺书**

1.本人郑重地承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。

2.本人在毕业论文（设计）中引用他人的观点和参考资料均加以注释和说明。

3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。

4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

毕业论文（设计）作者签名：

2019 年 6 月 10 日

致 谢

本文是在浙江大学杨赓老师和中国美术学院王菲、张英老师的指导下完成。在此毕业设计中“智能动作捕捉健身衣”部分的“需求分析”、“制作开发”、“总结布展”等各个阶段，美院的合作者均做出了重要贡献。她们工作的使得此健身衣在功能上，外形上，制作工艺上，展示效果上均有了显著提升。感谢美院同学的辛勤付出。

从由本科SRTP项目加入杨赓老师实验室以来已有两年有余，期间我接受了研究生般的科研上熏陶。感谢杨赓老师长久以来提供的学术上和人生上的指导、科研经费和设备上的支持。

感谢父母长久以来的学费上的支持，精神上的鼓励以及家庭的温暖。

毕业设计初期阶段我在国外交流学习。在我出国的这段时间里，感谢吕鸿昊、段鸿远程协助我完成自动引导运输车AGV以及可穿戴设备的测试、实验工作。也感谢庞高阳、周慧颖针对我的论文所提出的修改意见。

感谢毕业设计期间，陈飞宇、王方锦华、张浩对于我留学申请的指导。

最后感谢学院提供的和中国美术学院合作毕业设计的机会。

# 摘 要

根据世界卫生组织（WHO）的报告，世界范围内的老龄化人口将会在2015年至2050年内从9亿攀升至21亿。此外，根据中国第二次残疾人抽样调查人口显示，2006年中国肢体残疾的人数达到2412万。由于行动不便人群庞大的基数加上老龄人口的剧烈增长的速度，因此家庭护理机器人系统的开发便成为一项对社会有益的且十分紧迫的工作。

本文的工作集中于“人机交互系统”中的“可穿戴设备”和“智能机器人”两方面。其中可穿戴设备方面，本项目设计并制作了一款智能运动监测衣及附属设备。其主要功能有：1）监测人体上肢动作；2）血氧含量、心率；3）语音、触觉等多人机交互通道；4）与Android系统应用程序交流数据，并进行可视化显示。智能机器人方面，本项目对一款自动引导运输车进行了软件和控制层面的开发，实现了基于机器人操作系统ROS的上下位机通信以及机器人的建图、定位、路径规划功能。

第1章，阐述了研究背景与意义，综合列举并分析了现存的基于不同原理的人机交互设备的性能，同时也列举了国内外家庭护理机器人的研究方向，最后提出了本文的主要研究内容。

第2章，介绍了智能运动捕捉衣的功能和作用，对于其“机械结构”、“电路结构”、“软件系统”的原理以及设计和制作过程进行了具体的描述。

第3章，介绍了自动引导运输车的机械结构（主要是麦克纳姆轮的运动学特性）、电路结构（主要是控制器和驱动器）、软件架构（ROS操作系统），随后就“建图、定位、路径规划”的原理和具体实现方法进行了具体阐述。

第4章，对现阶段主要研究工作进行了总结，并提出了对未来工作的展望。

**关键词：**可穿戴设备，人机交互，自动引导运输车，机器人操作系统

# Abstract

According to a report of World Health Organization, aging population will rise from 900 million to 2.1 billion during the period of 2015 to 2050. Additionally, according to the second Chinese sample survey of disabled people, disabled population have reached 24.12 million since 2006 in China. Due to the huge amount of population of people with mobility disabilities in China and the rapid increase of aging population worldwide, researches related to assistive robotics have been a promising research topic.

The thesis focused on two topics of human-robot interaction—wearable devices and intelligent robotics. In the aspect of wearable devices, the thesis has proposed the designing and manufacturing procedure of an intelligent motion-captured suit. The main functions of this suit include: 1) upper limber motion capturing; 2) blood oxygen and heartrate monitoring; 3) multi-channel human-robot interaction based on speech signal and tactile sensation; 4) communication between Android application and central control unit. While, in another aspect of intelligent robotics, the thesis has proposed the designing of software and control system. To be specific, several functions, such as communication between host and slave port, mapping, localization and planning has been implemented, which were based on Robot Operating System on Linux.

In Chapter 1, the research background and significance of the dissertation were introduced. Then, different kinds of human-robot interaction devices were listed and introduced. Also, the main topics in the field of assistive robotics were introduced. Finally, the research contents of this dissertation were put forward.

In Chapter 2, the thesis proposed the main function of the intelligent motion-captured suit. The principle and technology of its mechanical structure, electrical structure and software system were proposed.

In Chapter 3, the thesis proposed the AGV’s mechanical structure, electrical structure and software system. The principle and implementation of AGV’s mapping, localization and planning functions were introduced further.

In Chapter 4, the research work of this thesis about human-robot interaction was summarized and future work was proposed.

**Keywords:** Wearable devices, Human-Robot interaction, Automated Guided Vehicle, Robot operating system.

# 目 录

[摘 要 I](#_Toc10020078)

[Abstract II](#_Toc10020079)

[目 录 III](#_Toc10020080)

[第1章 绪论 1](#_Toc10020081)

[1.1 研究背景与意义 1](#_Toc10020082)

[1.2 国内外研究现状 1](#_Toc10020083)

[1.2.1 人机协作系统的安全性和可靠性设计 1](#_Toc10020084)

[1.2.2 人-机器人交流通道的设计 2](#_Toc10020085)

[1.4 目前存在的主要问题和发展趋势 5](#_Toc10020086)

[1.5 论文的主要研究内容 6](#_Toc10020087)

[第2章 智能运动捕捉衣 7](#_Toc10020088)

[2.1 引言 7](#_Toc10020089)

[2.2 机械结构 8](#_Toc10020090)

[2.3 电路结构 9](#_Toc10020091)

[2.4 软件系统 11](#_Toc10020092)

[2.5 本章小结 12](#_Toc10020093)

[第3章 自动导航运输车 13](#_Toc10020094)

[3.1 引言 13](#_Toc10020095)

[3.2 车辆硬件系统 14](#_Toc10020096)

[3.2.1机械结构 14](#_Toc10020097)

[3.2.2电路结构 17](#_Toc10020098)

[3.3 机器人操作系统ROS 19](#_Toc10020099)

[3.3.1 ROS编译系统 19](#_Toc10020100)

[3.3.2串行通信模块 21](#_Toc10020101)

[3.3.3建图(Mapping)与定位(Localization)模块 21](#_Toc10020102)

[3.3.4路径规划(Planning)模块 24](#_Toc10020103)

[3.4 实验与测试 25](#_Toc10020104)

[3.5 本章小结 27](#_Toc10020105)

[第4章 总结与展望 29](#_Toc10020106)

[4.1 工作总结 29](#_Toc10020107)

[4.2 研究展望 29](#_Toc10020108)

[参考文献 30](#_Toc10020109)

[附件一：程序 32](#_Toc10020110)

[附件二：毕业设计成果 46](#_Toc10020111)

[毕业论文（设计）的进度安排 48](#_Toc10020112)

# **第1章 绪论**

**1.1 研究背景与意义**

微软创始人盖茨曾经提到：“目前机器人行业的发展同30年前电脑行业极为相似，我们完全能够想象机器人将会成为日常生活的一部分。”而如果想要让机器人更深入地介入人的生产和生活活动，就需要在人机协作领域取得突破，使得系统更加安全，人与机器人的交流通道更加畅通，同时在设计系统的过程中也要充分理解人的因素的作用。

以往工业机器人被安置在静态的环境中，由栅栏与工人隔离开，完成重量级的、单一重复的任务。而今年制造业产生的个性化、定制化、柔性化趋势使得这一生产模式收到了挑战[1]。在这种要求下，人与机器人之间的深度合作就不可避免，于是有一部分研究者就致力于工人与工业机器人的融合的研究。

另一方面，根据世界卫生组织（WHO）的报告，世界范围内的老龄化人口将会在2015年至2050年内从9亿攀升至21亿[2]。而医疗条件的增长速度跟不上老龄化的进程，又由于老年人口肢体行动能力退化，故缺乏看护的问题将会越来越严重且将广泛存在。此外，根据中国第二次残疾人抽样调查人口显示，2006年中国肢体残疾的人数达到2412万[3]。这一庞大的基数加上老龄人口的增长速度，使得开发能够填补医疗辅助领域空缺的家庭护理机器人系统成为一项对社会有益的工作。因此，又有一部分关于人机协作的研究是在服务机器人的背景下展开的。

人机协作机器人技术可以被应用于很多领域，除了工业机器人、辅助看护机器人外，还有抢险救援机器人、海洋太空探索机器人等领域。而本文以辅助看护机器人为主要研究和开发对象。

**1.2 国内外研究现状**

关于人机协作的研究内容主要有：1.人机协作系统的安全性和可靠性设计；2.机器人系统的智能提升；3.人-机器人交流通道的设计（包括用户发送指令的方式和反馈方式）。以上三点也是本次研究所关注的部分，后文将会分模块阐述具体的研究内容。

**1.2.1 人机协作系统的安全性和可靠性设计**

目前的工业机器人与工人协作过程缺乏安全性，且交互过程很不直观，因此机器人和工人目前还无法进行协作生产。安全是人机协作的前提条件。因此行业组织和政府为工业机器人制定了很多安全规范。涉及机器人的国家安全标准有工业环境用机器人安全要求（GB11291.1-2011）、机器人与机器人装备工业机器人的安全要求（GB11291.2-2013）、工业机器人安全实施规范（GB/T20867-2007）等，国际标准化组织也从1992年起就发布了机器人的安全性标准ISO10218并不断更新[4]。

除了行业组织制定的安全标准之外，国内外研究人员还从多个方面对人机交互的安全性课题进行了研究。例如Steneld等学者针对人机交互系统的特异性，总结了人机交互领域的通用性指标，包括了“感知、导航、管理、操作”等方面的内容。

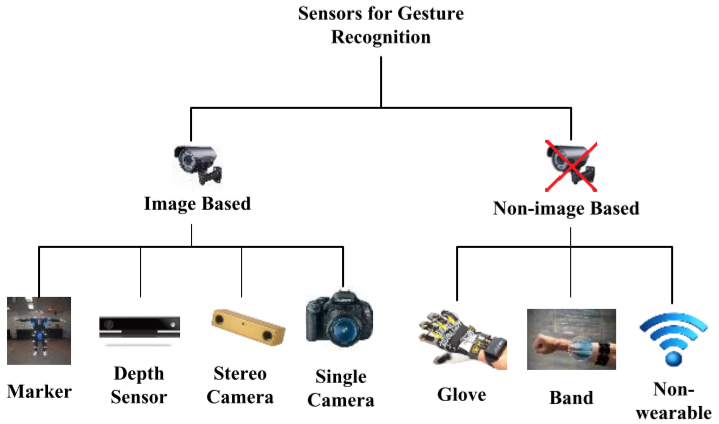
人-机器人协作可以分为两大类：非干涉性人机交互和干涉性人机交互。前者是指协作过程会偶尔产生物理碰撞的情况，后者是指人机协作要求人与机器人不断地进行身体接触。就非干涉性人机交互而言，除了电泄漏、高压油泄漏等机器人本身的机械故障以外，研究的重点落在如何使机器人应对突发情况这一问题上[5]，例如人的突然动作或机械臂的突然碰撞。安全性的解决方案通常包含机器人设计过程中的各个方面。总的来说可以概括为固有安全性和附加安全性。机器人机械结构方面的设计称为固有安全性设计，例如众多协作机器人产品都使用了轻量化设计，有些产品-例如YuMi-还在机械臂外包裹柔性材料以增加缓冲。附加安全性的设计改造偏向于软件和附加传感器这些方面，例如KUKA机器人的安全操作系统会实时监控机器人的位置、速度、加速度等参数。也有设计在机器人表面包裹柔性传感器、附着超声波传感器，以增加机器人的感知功能从而提高安全性。

对于干涉性人机协作系统来说，安全性的策略和要求都是和非干涉性的人机协作系统很不同的。例如Fanuc公司生产的IAD机械臂就需要人手作用于机械上的力用于输入指令。相比非干涉性的协作，此类协作方式相比远程操控拥有更高的运作效率，因为工人可以手把手地对机器人进行教学。然而改善安全性无可避免地会带来效率上的损失，因为必须要对机器人的运动速度和输出力进行限制，尤其是针对干涉性的人机协作系统。

**1.2.2 人-机器人交流通道的设计**

人机交互系统和人际交互系统有所不同，前者可能具有空间分离性和时间分离性（例如城市搜索救援机器人同人的交互就具有空间分离性，康复型轮椅同人的交互则是同时同地的）。于是这种特点就可以用于人机交互的分类。此外，从反馈信号的方面来看，有采用机械振动反馈（模拟触觉）[6]、视觉反馈（机器人系统数据可视化）、电刺激反馈等不同的反馈方式。本次设计采用的是视觉反馈方式。用视觉化呈现的方式在单位时间内向用户所呈现的信息量是最大的（相比于触觉和电刺激的方式），可以帮助用户快速判断机器人的状态，从而决定下一时刻的动作。

同时，人机交流通道（正向）根据所提取的人的不同行为而主要有分为两大类：基于动作识别技术的和基于语音识别技术的。本次研究主要针对基于动作的识别。而基于人动作的识别根据装置的不同又可以分为四大类，如图1.1所示：基于图像传感器的、基于手套系统的、基于绑带的、非穿戴式的。每一类都有各自的特点和优缺点，然而无论基于何种传感器，其信息处理的流程都是类似的，按照时间顺序排列如下：传感器产生信号、动作识别、动作追踪、动作分类。下面将分别介绍每一个步骤的具体内容。



**图 1.1 不同种类的动作识别传感器[7]**

如图1.1所示，在人机协作动作识别领域，基于图像的传感方案一共有四种：标识点法、深度传感器、立体摄像头、单摄像头。

对于标识点法来说，通常是使用光学摄像头进行的。当然，也有利用别的传感器来进行标点的，例如本文就使用了惯性传感器。通过追踪覆盖于人身体上的点关于参考点的相对偏移以及姿态，可以获得人的整体或者覆盖有标识点的部分肢体的位姿信息。

深度传感器是目前发展得最迅速的方案之一。优点是可以避免校准和打光问题对于动作识别的影响。其中比较常见的有微软公司生产的Kinect，基于庞大的开发者社群和开源资料，可以使得开发动作捕捉系统变得快捷。同时深度传感器可以和人体骨骼模型（动作识别步骤）相结合，使得动作分类步骤得以简化，即不必利用复杂的算法对捕捉到的人体姿态变化进行分类了。

对于立体摄像头来说，通常用的是双目视觉解决方案。然而计算量比较大且必须面对校准问题，这限制了这一方案的推广。

关于单摄像头方案，是从上世纪90年代开始产生的。一个显著的缺点就是视角受限导致的系统稳定性不佳。而最近又有人利用高速摄像机进行姿态识别，这样就可以捕捉快速运动情形下的姿态变化情况。

在非图像的传感方案中，包括手套系统、绑带系统和非穿戴式系统三大类。手套系统有着快速响应、精度高、受环境影响小的优点，比较适合用于需要高成功率的应用中。而其缺点是设备需要同人体接触，造成动作的不便。手套式的系统有主要有两个用途：其一是作为检测设备监测人手部的动作；其二是作为指令发生装置用于产生沟通或控制信号。此处属于第二种用途。如果以产生信号的连续与离散来分，手套系统又可以分为两类。而这两类都可以用于基于手套系统的动作识别应用中[8]。例如用手指的弯曲和伸直两种状态进行二进制编码控制底盘移动就属于离散系统。而如果是在“实时监测手指的弯曲程度进而映射到机器手”这样的应用中，手套就属于连续系统。上述两种系统在信号处理和识别的时候用到的方法很不一样，总的来说连续系统的信号处理和分类会更复杂。手套式系统可以用在不同的领域，因此信号处理方法、设计方法在不同应用之间相差非常大，而研究者Sturman和Zeltzer提出了一个概念，即不同应用就是不同的将手部动作的映射（映射至符号或者命令等），可以用统一的观点来看待不同的基于手套系统的应用[9]。

绑带系统也是基于可穿戴设备的，这种类型的传感器捕捉面积大，常用于捕捉肌肉电信号以控制设备。其优缺点与手套系统类似。

非穿戴式设备通常基于射频和电磁波发生和接收装置，通过人体反射的电磁波来进行动作捕捉。目前这种方法相对而言尚未成熟，主要的缺陷在于分辨率较低。然而优势在于可以穿透障碍物进行人体动作的捕捉。

以上便是传感器的分类。而为了得到人体动作姿态，需要对于传感器信号的进行处理。通常的处理步骤可以分为以下三个阶段：（1）动作识别（2）动作追踪（3）动作分类。下面将详细介绍三个步骤的作用：

（1）动作识别的技术主要有特征提取、人体模型、学习算法三大类。不同的动作识别技术适用于不同的传感器，同时也拥有各自的优点和缺点。针对单摄像头系统，通常可以采用视觉特征提取算法和学习算法。对于深度摄像头系统（例如基于Kinect的系统）来说，人体骨骼模型是最佳的选择。而绑带或手套系统常常需要进行数字滤波，处理过后的信号再可以用学习算法进行处理。由于非穿戴式设备易受空间电磁场干扰的特性，在进行动作识别之前也需要用高级的滤波算法进行处理。

（2）动作追踪步骤主要是为了前后数据间的变化，从而确定人姿势的动态变化。而对于静态的姿态识别应用，这一步可以略过。常用的动作追踪算法有单假设追踪（包括扩展卡尔曼滤波、平均值偏移等）、多假设追踪（包括扩展粒子滤波等）、扩充模型的追踪、检测追踪算法等。一般来说对于短时间、单目标的动作追踪常用的是单假设方法。而多假设方法和扩充模型法分别用于多目标和长时间的动作追踪。最后的检测追踪算法是一种学习算法，适用于复杂环境。

（3）动作分类是整个动作识别工作流程中最重要的一环，常用的算法有：K-最邻近算法、隐马尔可夫模型、支持向量机、集成方法、动态时间规整、深度学习及各种机器学习算法。每一种算法都有其各自的优点和缺点。最近由于计算设备性能的提升，深度学习算法获得了广泛应用。同时，将不同算法结合使用以获取各自的优势的做法也十分流行。

**1.4 目前存在的主要问题和发展趋势**

就可穿戴设备中的手套式系统来说，最主要的缺点就在于对使用者的束缚和对触觉的剥夺。戴上手套之后人用手从事精细化工作的能力会减弱，同时由于手套有一定的重量，也会对人的行为灵活性造成负面影响。另一些缺点是材料的疲劳性、相对较高的制造成本以及使用前的校准要求等[9]。可穿戴设备的传感方案一般都具有此类缺点，因此未来的发展趋势是针对此类缺点进行改进。

就基于深度传感器的方案而言，以大公司开发的硬件产品为基础，开发人员形成社群互相交流、迭代代码，这一模式使得此类方案发展势头强劲。非穿戴式传感器的技术虽然目前还不成熟，但有广阔的发展前景[7]。

就手套系统而言，有逐渐轻量化的趋势，有的手套甚至摒弃了布料的支撑而直接将传感器紧贴在手上[10]。有的学者试图提高手套系统的耐用性。也有研究者试图降低手套的制造成本，或者将传感材料直接编织在手套上而作为手套机械结构的一个部分[11]。还有的试图去除手套使用前的校准步骤。以“围绕手套系统所存在的问题”进行的改进的工作居多。

**1.5 论文的主要研究内容**

此次毕业设计将设计一个用可穿戴式设备控制家庭护理机器人的人机协作系统。整个人机协作系统的方案架构包括“可穿戴式设备（运动捕捉衣）、机器人操作系统ROS（自动导航车）”两个部分的内容。

可穿戴设备用于捕捉人体的姿态，同时还配备有监测姿势、语音交互、人体生理指标的功能。希望通过此系统，向用户提供健康、运动指导建议。同时可穿戴式设备可以与自动引导车相结合，构成具备多通道人机交互功能的人-机器人系统。

本次设计采用基于标识法的方案来检测人的四肢、躯干等部位的运动姿态，将在使用者身上布置弯曲传感器。使用前进行校准调试，通过中心计算单元的数据处理，将会输出传感器安装部位的弯曲信息。以正确运动姿势的动作参数为基准，此系统可以实现纠正人体不正确的运动姿势的功能。此外系统还将针对使用者的部分生理指标进行监测。健康监测设备通常会监测的生理指标有：血液温度、心率、呼吸频率、血氧含量、血压[12]。此次设计将通过反射式光电容积法，对用户的血氧浓度进行监测和心率等身体指标进行监测。其原理可参考文献[13]。

自动导航车的姿态、位置等信息将被实时地反馈到计算平台，通过计算单元的处理后形成反馈控制信号，使AGV具备自动寻路、避障运行的功能。在ROS计算平台上运行Linux操作系统，其上又运行有ROS系统，用以实现“传感器数据融合”、“运动学解算”、“实时定位和建图（SLAM）”、“路径规划”、“可视化数据”、“机器人坐标管理”、“通信”等应用层功能。通过ROS系统的节点（线程）计算通信网络，可以实现此看护机器人的自动化运行、数据通信、视觉反馈等功能。ROS平台通过应用层的计算后会将运动指令通过串口发送给下位机，即辅助机器人系统的控制器。机器人系统将根据指令做出相应的动作。于此同时，机器人状态的变化通过相应的激光雷达、里程计、编码器、摄像头等传感器进行捕捉后反馈回ROS平台，数据将被传送给用户端并以可视化的形式（模型和地图）呈现给操作者，而操作者将通过这些视觉信息进行下一时刻的动作决策。

# **第2章 智能运动捕捉衣**

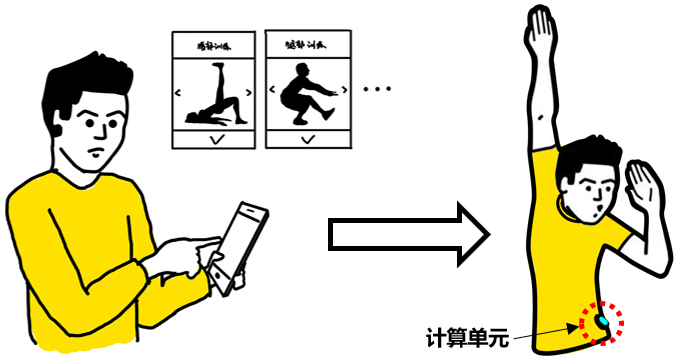
**2.1 引言**

非专业运动人士在运动的过程中无法确定某一运动动作标准与否，因此需要一款智能可穿戴设备对人的运动姿势进行监测。针对这一需求，设计了一款用于捕捉人体姿态的智能运动衣。

本次设计的智能运动捕捉衣是一款能够捕捉人体上肢（肘部、肩关节）弯曲角度的仪器。该仪器同时具备多维度人机交互通道，例如：人机语音交互功能（包括语音播报和语音识别）、震动反馈功能，同时用户可以通过专用Android平台APP与设备进行交互。APP将在用户运动的同时播放正确姿势画面。

此设备可以捕捉用户的动作并转换为电信号，数据处理平台将信号进行数字滤波、动作检测追踪、动作识别、数据整合等操作后，通过串口通信技术把数据发送给ROS硬件平台。同时可以捕捉人体的动作，达到能够控制双机械臂机器人（YuMi）双臂的操作和自动引导车（AGV）的运动的功能。

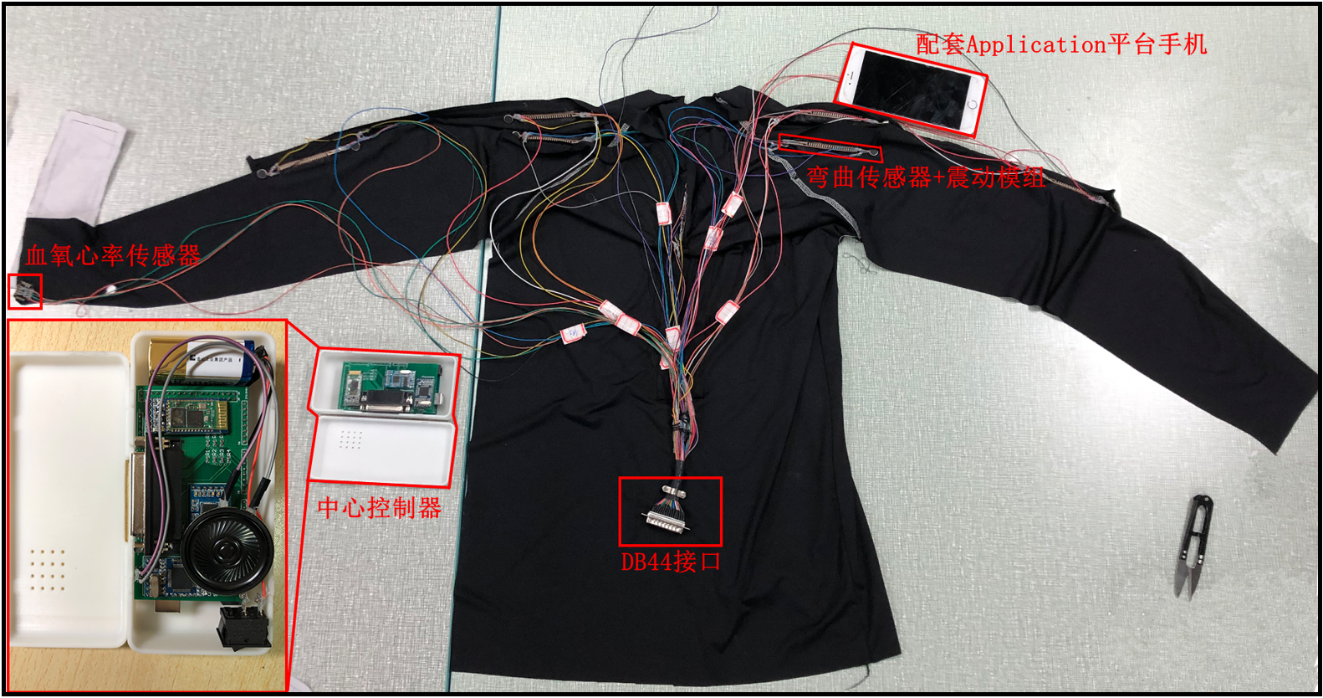
本章将从机械结构（衣服、线路以及模块容纳装置等）、电路结构（针对中心计算设备Arduino Mega设计的专用接口电路PCB板）、软件系统（实现人机交互、姿态识别和上下位机通信的功能），三个方面进行阐述。



**图 2.1 智能运动监测衣使用步骤**

如图2.1所示，使用者在手机应用软件上选取锻炼动作。运动监测系统通过弯曲传感器监测四肢和躯干的动作并与标准动作进行对比。若使用者产生动作不标准的情况，系统将通过语音提醒和震动反馈纠正。

**2.2 机械结构**



**图2.2 智能健身衣的机械结构图**

智能动作捕捉健身衣的设计参考普通健身衣的版型，用服装印花的工艺进行图案制作，其领口袖口采用不同颜色不同材质的布料增添衣服的层次感。

七条弯曲传感器分别置于左右手的“肩峰角”、“肩胛颈”、“肘关节外部”以及“背部胸椎”处。用于检测“抬肩”、“前合掌”、“屈臂”、“挺胸”这四个动作的标准程度。同时弯曲传感器两端安装扁平震动马达，用于发现动作不达标时的震动提醒。将所有弯曲传感器、震动马达和心率检测模块的接口连接到26AWG规格的铜芯控制电线束上，并将电线束汇集到DB44公端口上。为了保证元器件与控制电线连接处的牢固性（事实上在增加热熔胶步骤之前，电线连接处非常薄弱，运动过程中容易断裂），除了锡焊工艺外，此项目中还增加了热熔胶固定的步骤。

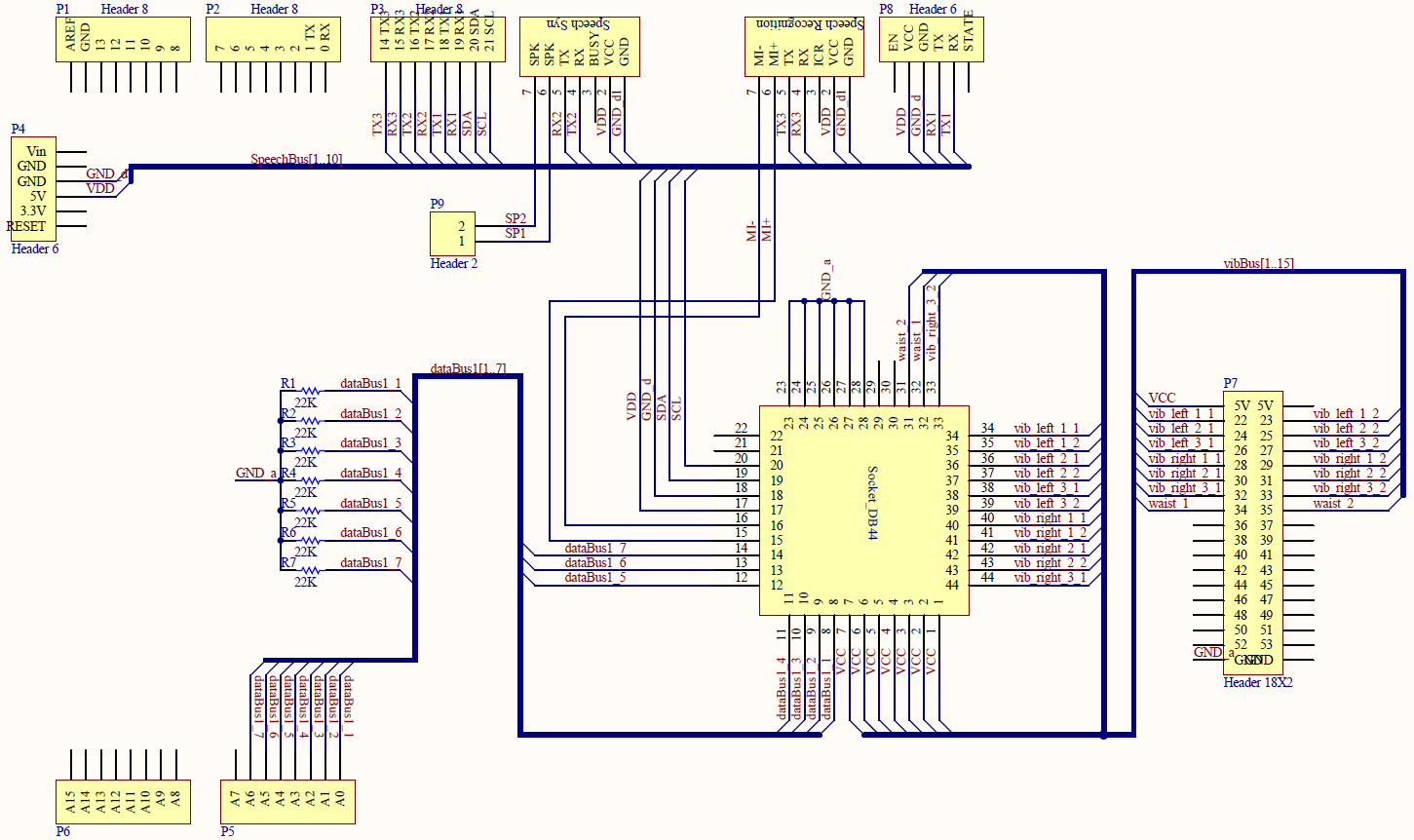
控制电线束在衣服表面走线方式描述如下：从人肢体末端到躯干部位，采用“支流汇聚到干流”的走线方式，将小股电线在背部胸椎位置汇聚成大股。电线表面用（与健身衣）相同材质的布料覆盖并缝合。最终使得可以通过DB44接口把衣服上附着的电子元器件整体同中心计算设备连接，构成动作、心率检测和反馈电路系统的一部分。

中心控制和计算单元（Arduino Mega及其转接板）、电池模块均盛放于塑料盒中。塑料盒在运动过程中可以放置在后背腰部布袋中。塑料盒外观采用由Robert McNeel & Associates公司开发的3D CAD软件Rhino设计，导出STL格式的文件后进行3D打印制作。后期钳工进行掏孔、打磨、固定、喷漆等加工流程，使盒子外形更美观。

**2.3 电路结构**

采用Arduino Mega 2560作为中心计算单元，用于实现“电压信号的模数转换”、“I2C通信（同心率血氧检测模块之间）”、“UART串口通信（同语音识别、语音播报、蓝牙传输模块之间）”、“软件流程控制”、“运动检测和标准化判断”几项功能。选用Arduino Mega2560的主要原因在于其接口数量和种类符合项目要求：数字输入输出口0~53、模拟输入输出口0~15用于连接弯曲传感器和震动马达；I2C接口用于连接心率血氧传感器；USART1~3串口分别用于连接，蓝牙模块、语音播报模块、语音识别模块，接口空间浪费率很小。同时其计算速度、存储资源足够实现预设计的功能，同时大量开源函数包赋予此模块的快速开发的特性也非常适合此次项目。

而Arduino Mega作为通用开发板，它与周围电路的连接需要用到大量导线，为了使装置简洁可靠，需要根据Arduino Mega尺寸和接口功能制作专用的转接板。其电路原理图如下所示。

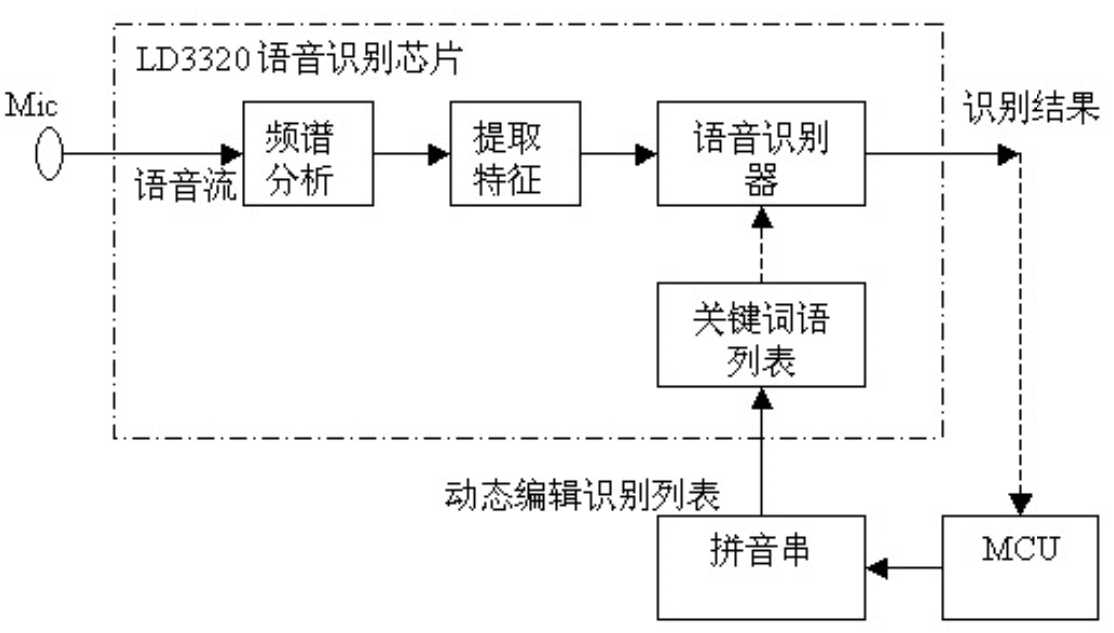


**图2.3 智能运动监测衣转接板电路原理图**

血氧和心率检测模块采用MAX30102功能电路模块，其原理是红光和红外光LED发出的光线穿过皮肤。血液中的氧合血红蛋白和血红蛋白吸收特定波长的光线并反射，进而通过光电转化元器件检测反射光线强度，经过噪声抑制电路和通信子模块后可以通过I2C接口向外界输出心率和血氧的数值。

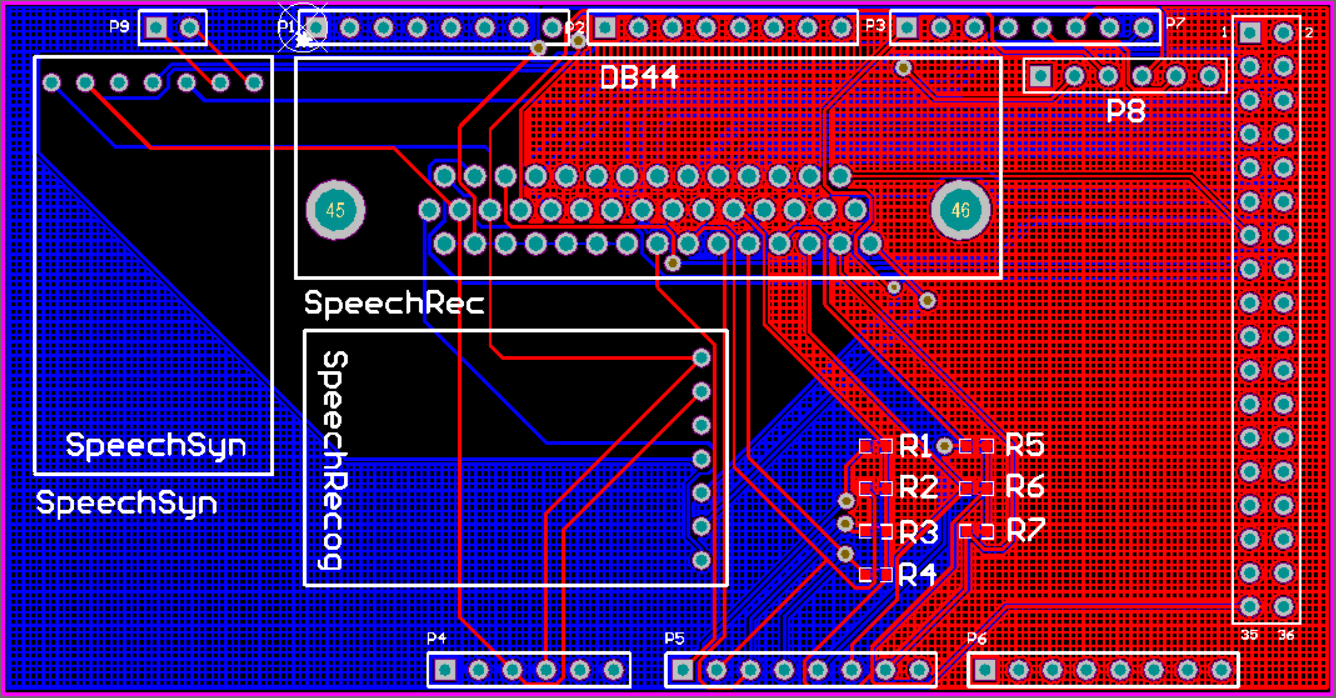
蓝牙模块采用HC05（搭载CSR蓝牙芯片）。此模块有着价格低、开发迅速、集成度高的优势，设置为蓝牙从机的设备可以同其他蓝牙主机、手机或电脑等终端设备连接，非常适合实现智能健身衣项目的蓝牙功能需求。

语音播报的核心功能模块是SYN6658型号芯片及周围UART通信、供电、晶振等功能电路。模块内置语音库且使用串口与外界通信，用于合成相应的语音信号。板载有D类功放能够直接驱动扬声器。本模块同Arduino Mega进行通信，用于播放提示语音信号。



**图 2.4 LD3320语音识别芯片数据处理流程图[14]**

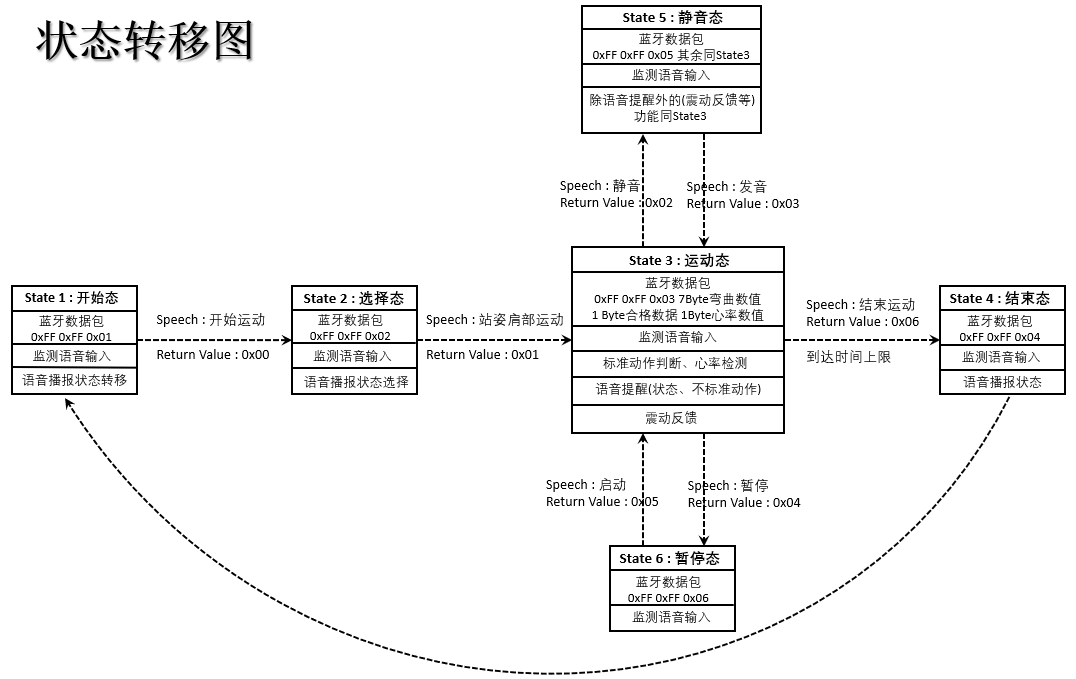
语音识别模块LP-ICR的核心功能子模块是LD3320芯片及周围UART通信、供电、晶振等功能电路。其中LD3320芯片使用了非特定人的语音识别技术。研究人员通过提取“元语音”之间的差异化特征，再将语音统计学模型用硬件电路的方式进行固化，从而制造出了LD3320芯片。其工作方式值得注意：麦克风将声音信号转化为电压信号后传输给芯片，首先进行A/D转换，再通过“频谱转换”并对频域信号进行“特征提取”，进而和“关键字列表（用户事先通过串口输入拼音，并由芯片动态地构造此列表）”中的项目进行一一对比，选择最优匹配的一条作为识别结果。其工作参数还可以通过配置相应的寄存器进行更改（如语音信号的有效性标准、噪声剔除标准等）。



**图 2.4 智能运动监测衣PCB原理图**

如前文所述，需要根据Arduino Mega 2560的接口定制相应尺寸的转接板，上图所示的PCB图展示了将心率检测模块接口、蓝牙模块、语音识别模块、语音播报模块集成在“转接板”上的排布和走线。使用时只需要将转接板插入中心计算控制模块的相应接口，硬件电路就可以正常使用。

**2.4 软件系统**



**图 2.5 软件流程状态转移图**

在Arduino Mega 2560、MAX30102、HC05、SYN6658、LD3320等逻辑功能电路模块，传感器模块（弯曲传感器分压电路），反馈模块（震动马达）的硬件基础上，实现了六大功能：肘部肩部动作捕捉、语音提醒、震动反馈、语音识别、通过Bluetooth与Android手机应用通信、血氧心率数据采集。程序设计面向从“开始运动”到“运动结束”的一个完整的锻炼周期，相应地采用了流程控制的设计思路。

流程控制采用了状态机（Finite State Machine）的设计思想。有限状态机是一种描述有限数量的状态以及状态之间转移的行为和条件的模型。并且此项目中的软件模型属于Moore型有限状态机，即状态与输入信号有关，而输出信号只与当前状态有关。如上图所示，程序规定了六种状态，分别为“State1：开始态”、“State2：选择态”、“State3：运动态”、“State4：结束态”、“State5：静音态”、“State6：暂停态”。如果在程序执行时的某一瞬间进行分析，则程序必然处于六种状态中的其中一种。状态之间的切换的条件是用户输入的语音信号（识别后的返回值）。在不同状态下，软件实现的功能有差异，例如：系统在“运动态”的时候检测人体动作信号和心率血氧信号，做出标准与否的判断后进行语音和振动马达反馈，同时通过蓝牙将运动数据发送给Android系统；而在“静音态”的时候，不仅是通过蓝牙发送的数据包有所不同，而且取消了语音播报功能。

**2.5 本章小结**

本章是关于人机交互可穿戴设备的研究，开发了一整套监测人体（上肢）运动的装置。主要工作集中在“需求分析及功能定位”、“服装及附属设备机械结构的设计与实现”、“逻辑功能电路及传感反馈子系统的设计与实现”、“软件架构设计与实现”四个方面。

（1）在产品功能定位阶段，结合“锻炼人群”和“需要的运动监测的人群”的需求，设计了“语音识别”、“震动反馈”、“语音反馈”、“Android手机应用交互”这些人机交互通道。为了展现监测人体运动的效果，本项目选用了“站姿肩部深层激活运动”动作作为标准化的检测动作。

（2）在机械设计阶段，主要解决的问题是“传感器”、“震动马达”、“电线”、“嵌入式计算单元”的布置位置和固定方式。七条弯曲传感器分别置于左右手的“肩峰角”、“肩胛颈”、“肘关节外部”以及“背部胸椎”处，用于检测动作的标准程度。两端扁平震动马达用于震动提醒。控制电线束采用“支流汇聚到干流”的走线方式。三者均用布条包裹，针线固定。同时，电线连接处还使用了热熔胶进行固定。嵌入式控制和计算单元、电池模块均盛放于3D打印制作并进行掏孔、打磨、固定、喷漆加工的塑料盒中。塑料盒在运动过程中可以放置在后背腰部布袋中。

（3）在电路设计阶段，主要搭建了嵌入式系统硬件。硬件主要用于实现“信号模数转换”、“I2C通信”、“UART串口通信”、“软件流程控制”、“运动检测和标准化判断”几项功能。I/O口连接“弯曲传感器”和“震动马达”；I2C接口连接“心率血氧传感器”；USART1~3串口分别连接“蓝牙模块”、“语音播报模块”、“语音识别模块”。

（4）在软件设计阶段，面向从“开始运动”到“运动结束”的一个完整的锻炼周期，整个流程控制采用了状态机（Finite State Machine）的设计思想。在上述硬件基础上，实现了六大功能：肘部肩部动作捕捉、语音提醒、震动反馈、语音识别、通过Bluetooth与Android手机应用通信、血氧心率数据采集。

# **第3章 自动导航运输车**

**3.1 引言**

ROS是一种编写机器人应用程序的灵活的软件框架，它拥有一系列软件工具、功能库，这使得机器人功能开发更便捷[15]。同时，由于整个架构采用模块化设计，于是开发者之间就可以分享彼此的代码，而使得开发出的系统结合了各个模块的优势。例如：建图、导航、计算机视觉、数据融合等模块都可以进行组合替换使用。如图所示，为了实现底盘AGV部分的自动化运行，需要6个ROS线程（矩形部分）通过发布相应消息进行相互通信并组成线程网络。



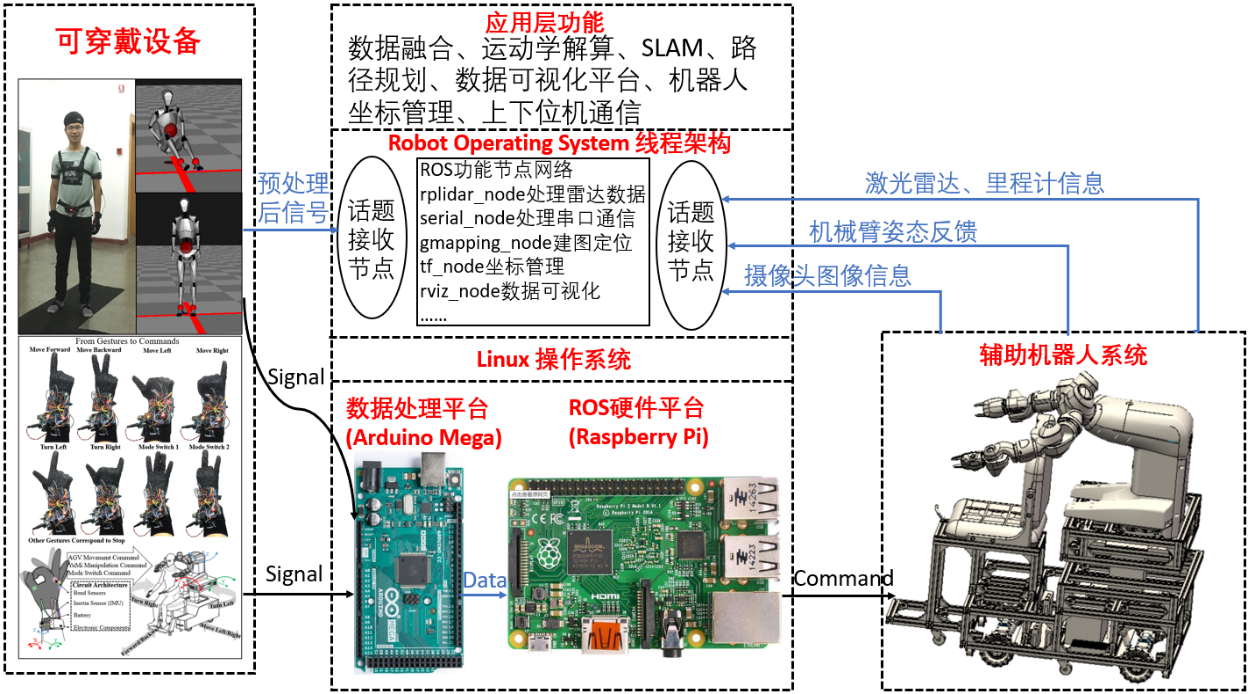
**图3.1 ROS线程架构的设计（AGV部分）**

其中建图（Mapping）模块、定位（Localization）模块、路径规划（Planning）模块、坐标管理模块能够构建出全局和局部地图，进而实现定位和导航功能后计算出机器人运动学参数。通过串口通信模块能够将运动学参数（如：速度）以“上位机-AGV主控制器应用层协议”规定的格式发送给机器人系统。在ROS系统中，建图模块常用的选择有采用粒子滤波（PF）的gmapping模组、基于优化方法的hector模组以及Cartographer模组等[15]。事实上gmapping模组是基于Rao-Blackwellized粒子滤波算法编写的[16]，此算法的具体阐释见机器人操作系统ROS一节。现代SLAM系统的结构通常分为两个部分：基于传感器的处理模块（前端）和最大后验估算模块（后端）[17]。其中，前端模块的功能由激光雷达系统实现，ROS系统中的模组主要承担后端模块的功能。定位导航模块通常使用自适应蒙特卡罗方法（AMCL），此算法的具体阐释见机器人操作系统ROS一节。路径规划模块通常由“全局路径规划”和“局部路径规划”两部分组成，其中全局路径规划采用的通常是Dijkstra算法或A\*算法，其中Dijkstra算法的具体阐释见机器人操作系统ROS一节；而局部路径规划可以采用DWA、Trajectory、TEB等算法，DWA算法的具体阐释见机器人操作系统ROS一节。

此外，为了控制人机协作系统中的另一个设备YuMi机器人，并优化用户体验，还可以构建另一个类似的ROS线程架构。其功能是关于“发送YuMi控制指令”、“接收摄像头信号”、“运动学解算”、“数据可视化”等内容的。如果将两个架构结合起来后可以真正实现机器人“系统”的自动化运行和人机交互控制。针对AGV和YuMi的两个架构相对独立，而本此设计只针对AGV的部分进行开发。

**3.2 车辆硬件系统**

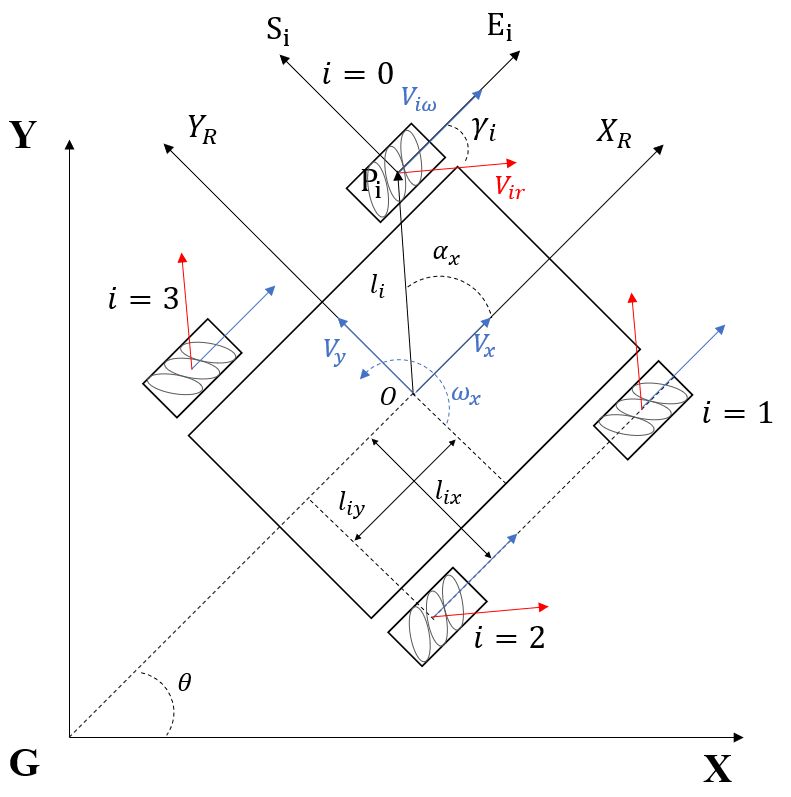
**3.2.1机械结构**



**图3.2 人机交互系统架构**

家庭看护辅助机器人系统是由ABB公司的YuMi型机器人和为家庭辅助机器人系统定制的自动引导车（AGV）两大模块组成的，见图3.2右下角。机械结构方面，移动平台AGV设计有四个麦克纳姆全向移动轮。由于不会有转弯半径的限制，所以机器人系统可以在相对狭小的空间进行全向移动。而YuMi和AGV的连接采用升降机构固定连接方式，可以让YuMi获得更大的工作范围。

为了实现AGV的全向移动需要对AGV的主控制器（相对于ROS硬件平台的下位机，安装在AGV机械框架中）进行开发：首先需要得到基于麦克纳姆轮的AGV的运动学方程；其次，需要获得电机及其驱动器的参数和通信协议，以便AGV主控制器输出正确的电机控制信号（直接控制电机或麦克纳姆轮的转速），从而控制AGV的全向移动。本小节对机械结构的分析主要集中在AGV的麦克纳姆全向轮运动学方程的推导上，关于电机控制的内容放在下一小节中。



**图 3.3 AGV麦克纳姆轮运动学分析参数图（参考[18]）**

：AGV固定坐标系；

：麦克纳姆轮坐标系；

：AGV在惯性系X-G-Y下的位置和朝向；

：AGV在平面上的移动速度和转动速度；

：AGV的轮间距的一半和轮子到AGV中心的距离；

：麦克纳姆轮转速，麦克纳姆轮上的滚轴转速，麦克纳姆轮与车身夹角；

在忽略轮子与地面之间的滑动的前提下，四麦克纳姆轮移动系统的逆运动学[18]:

(3-1)

其中为四麦克纳姆轮小车逆运动学（即从四轮小车的实际平面速度反推四个麦克纳姆轮的转速）的雅可比矩阵，于是可以得到：

(3-2)

本项目所用的自动引导运输车的麦克纳姆轮参数如下表3.1所示：

**表3.1 麦克纳姆轮实际参数**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 0 |  |  |  |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

因此逆运动学方程为（从AGV整体的运动速度变换到四个麦克纳姆轮的旋转速度）：

(3-3)

于是可以推导出正运动学方程为：

(3-4)

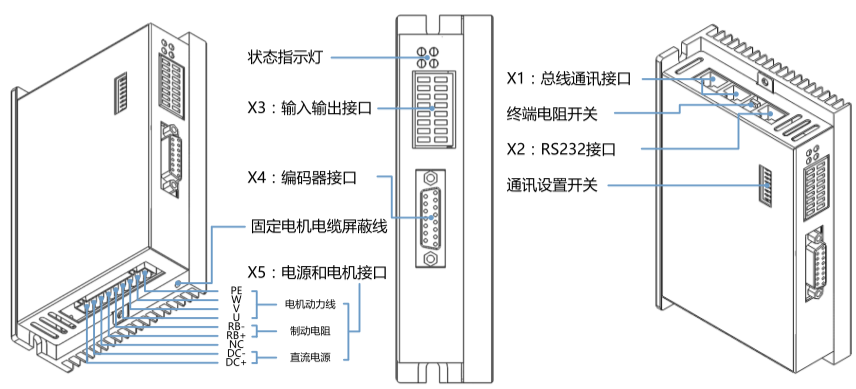
运动学雅可比矩阵中的待定参数涉及AGV的尺寸结构（轴、轴的半程轮间距），这一部分的参数可以参考AGV配套的设计开发手册。根据运动学方程可以获得AGV整体“前进、后退、左移、右移”四个平移速度和“左旋、右旋”两个旋转速度下，所对应的四个麦克纳姆轮的角速度。以上内容是对AGV主控制器进行有关“电机控制”的设计的基础。

**3.2.2电路结构**

控制系统方面，AGV的主控板使用了STM32单片机作为主计算单元，且电机的控制参数已经调整好并留有接口（只需要写值入特定内存地址），因此上位机通过串口发送规定格式的数据到主控板单片机，经过数据处理后，主控板再与四个轮子的驱动器进行通信，由麦克纳姆轮调整不同转速，从而实现AGV全向移动控制。AGV上搭载的YuMi机器人则通过I/O口接收上位机发送的通讯指令从而进行相应的动作。

由于AGV装载电源是DC24V的，控制器额定电压DC3.3V，CAN总线子系统额定电压DC5V，因此AGV车载控制器嵌入了电压转换模块。进而可以将控制器与驱动器各模块相连。

本项目的AGV所采用的是无刷直流伺服电机SMC60S-0020-30AAK（上海步科自动化股份有限公司），驱动器型号为FD123-CA-000，如下图3.4所示：

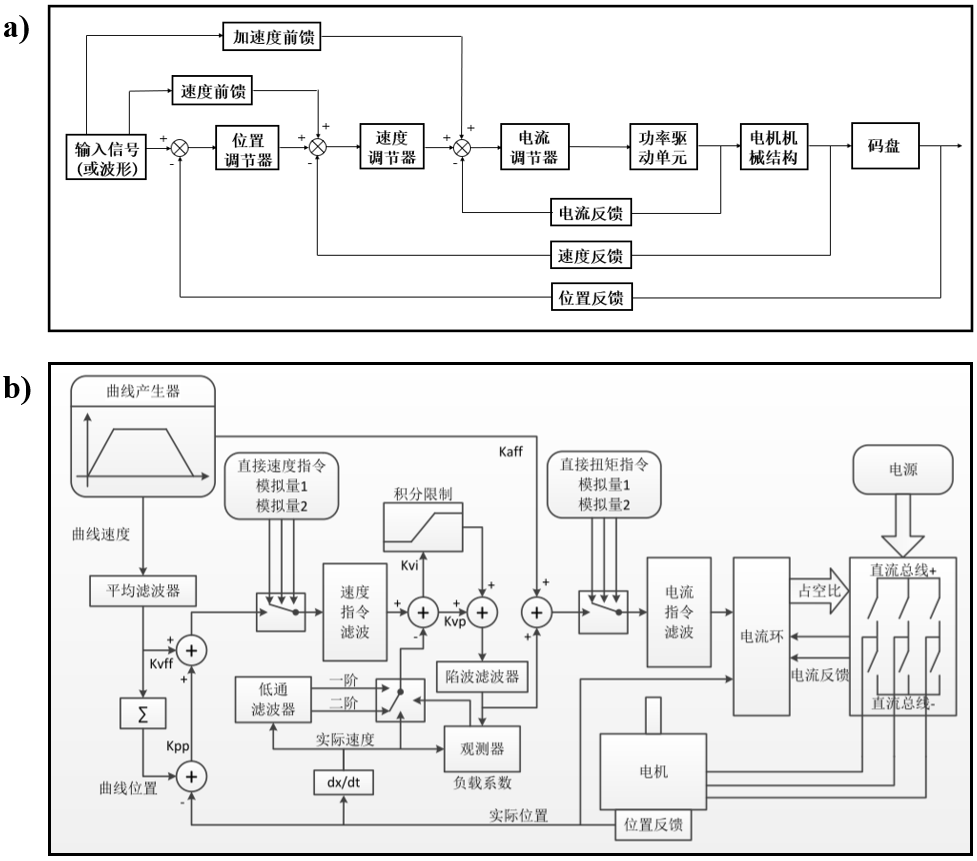


**图3.4 FD123-CA-000驱动器接口图示[19]**

电机驱动器需要与AGV中央控制器（STM32F405RGT6）进行通信，以接收速度参考信号。同时AGV的四个麦克纳姆轮需要由独立的伺服电机进行驱动，因此驱动器之间也需要相互通信。

驱动器的内置通信方式有多种选择（RS232、RS485、CANopen）。本项目的通信方式描述如下：AGV控制器作为主站端，而四个驱动器作为CAN从站端依次采用“级联”的方式相互连接。根据手册要求：主站和末尾从站之间需要接120的终端电阻（采用的是驱动器内置电阻）。

如图3.4所示，此型号驱动器的通信接口有X1：RS232、总线通信接口X2：CANopen，此项目所采取的通信方式是基于CANopen的。CAN规范定义了OSI模型中底层（物理层与数据链路层）的协议。本项目物理层的连接采用了双绞信号线，而设置数据链路层协议是为了防止数据传输过程中出现差错。由上一节AGV的运动分析可知，为了能够实现全向移动且速度可控，需要准确控制四个独立麦克纳姆轮的转速。因此驱动器选用“立即速度”模式，接收编码器的反馈信号（X4）。另外，X3输入输出接口连接的是抱闸线缆和电源，电机和驱动器的供电接口用的是X5。



**图 3.5 a)****SMC60S-0020-30AAK直流伺服电机“电流环”、“速度环”、“位置环”三环控制框图;**

**b)FD123-CA-000型控制器框图细节[19];**

如上图a)所示，直流伺服电机的控制包含三个闭环：“电流环”、“速度环”、“位置环”。首先，“电流环”的输入是“速度环”的输出加上“输入波形”的前馈信号，而其输出是电机的相电流，反馈信号来自于电机内部的霍尔元件。其次，“速度环”的输入是“位置环”的输出加上“输入波形”的前馈信号，其输出构成驱动参考电流，其反馈信号来自于电机内部码盘（位移的微分）。再其次，最外环“位置环”的参考信号来自于输入波形，其输出信号构成“速度环”的参考信号，反馈信号同样来自于电机内部码盘。这张图表达了直流伺服电机控制的理论模型，通用于不同品牌型号的电机。但此图的缺点在于，对于一些细节进行了简化处理，例如前馈反馈环节滤波器的选择、PID参数、积分单元的上限等。

针对FD123-CA-000型控制器内部参数，还可以画出如上图b)所示的控制框图，用于指导控制器调参。调试过程主要调整的是“速度环”和“位置环”的参数，而“电流环”通常默认为所选电机的最佳参数，故不需要调整。

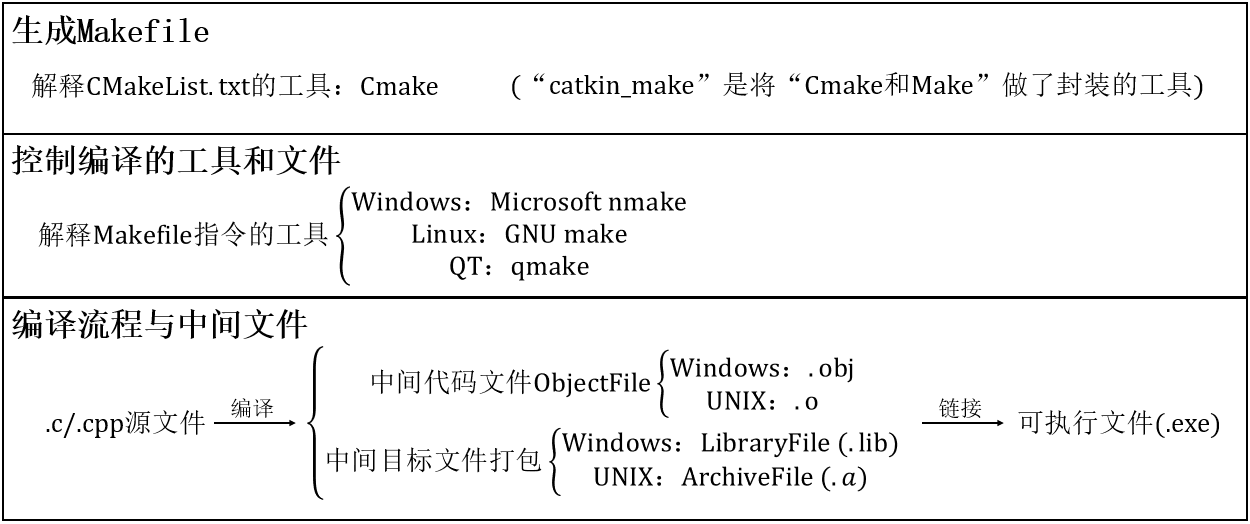
伺服控制器的“速度环”通常采用PI控制（此控制器也不例外），调整过程中需要依次确定以下参数：①速度环比例增益；②速度环积分增益（分为高精度和低精度）；③速度反馈滤波参数；④速度反馈模式；⑤输出滤波；⑥速度环积分限制。所有参数均保存于内存中，调整时只需要往特定内存地址写入数据，此步骤可以参考伺服电机使用手册。

伺服控制器的“位置环”通常采用比例P控制，且存在对“速度环”和“电流环”的前馈信号，调整过程中需要依次确定以下参数：①位置环比例增益；②速度前馈和加速度前馈；③平均滤波；④最大跟随误差。

由于本项目执行机构不需要运行到固定位置，只需要四个麦克纳姆轮能按照给定速度旋转，因此选用电机控制器的“立即速度”模式。在三闭环调整完成并选定立即速度模式后，上位机只需要往地址为60FF0020的内存中写入格式为Integer32的目标速度数据，电机就会按照给定速度旋转。这便是伺服电机留给STM32控制器的控制接口。

**3.3 机器人操作系统ROS**

**3.3.1 ROS编译系统**



**图 3.6 跨平台编译工具与典型编译过程**

ROS系统自带了Linux平台的编译工具“catkin\_make”，此工具集合了Cmake和Make工具的功能，其作用在于实现跨平台的编译。典型的编译流程见上图所示。

不同编程语言写成的源文件（ROS系统上可以用Python和C/C++）经过编译器（如GNU Compiler Collection）的处理后先生成中间文件，下一步经过链接操作生成可执行文件。其中的编译命令由控制台进行输入。然而当工程较复杂时（如头文件依赖项较多），为了简化手动输入编译指令的操作，于是引入了“控制编译的工具”（如GNU make配合Makefile文件），不同平台的控制编译工具和Makefile不同，因此会造成无法跨平台使用的情况。而为了让程序能够进行跨平台编译，又引入了Cmake工具，配合CMakeList.txt文件，可以生成不同平台的Makefile文件从而控制编译的流程。

**表3.2 ROS—Cmake (Catkin) 编译工具配置文件CMakeList.txt程序结构**

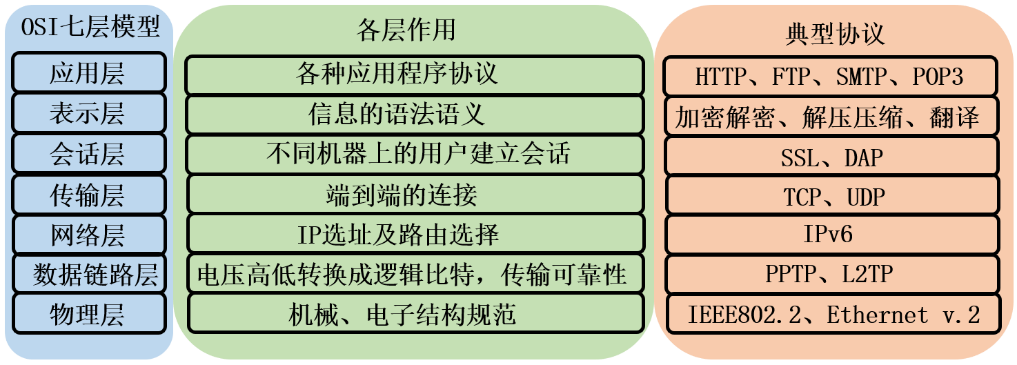


在创建ROS系统工程的时候，与在Windows平台Visual Studio IDE下编写程序不同，除了编写C语言源文件外，还需要编写为配置“编译功能”服务的CMakeList文件。文件的结构如上表所示，描述了编译过程所依赖的其他ROS包文件、编译ROS包生成的消息/服务/动作等文件、编译过后生成的库文件/可执行文件等信息。有时ROS工程会出现编译错误的问题，除源文件语法错误外，CMakeList文件出错的概率也很大。

**3.3.2串行通信模块**

在OSI七层网络模型的框架下，对（ROS系统与AGV控制器STM32之间的）串行通信进行描述。物理层采用铝箔屏蔽双绞线，两端接口分别为USB和九针串口，配备有CH340芯片进行协议之间的转换。数据链路层采用CRC循环冗余校验码的方式来确保上位机与AGV底盘数据通信的正确性。应用层的协议的作用在于解释ROS系统向AGV下位机发送的数据包的含义。

OSI七层网络模型“各层级的作用”、“每层的典型协议”见下图：

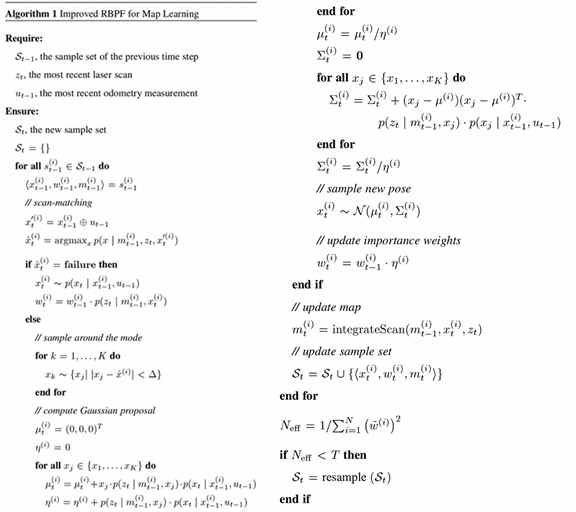


**图3.7 OSI七层网络模型**

此项目的“数据链路层”协议具体实现方式是：上位机将控制指令以数据帧的方式封存在结构体当中，并通过查表法计算出控制指令CRC16的值，附加在控制帧的末尾。下位机接收控制帧后进行同样的校验操作并对比校验值。若校验值相符，则说明上下位机数据通信无差错。而“应用层”则以C语言结构体的形式实现了“上位机-AGV主控器”通信协议中定义的数据帧。

**3.3.3建图(Mapping)与定位(Localization)模块**

硬件方面采用思岚公司的RPLIDAR A2型号激光雷达产品。激光雷达比较重要的参数有两项：最大有效扫描半径为16米，最大支持扫描频率为10赫兹。正常工作环境为无强光源照射的室内环境，符合本项目的室内移动机器人的要求。激光雷达接口为串口（UART），因此需要串口-USB转换器以实现与PC上位机的连接。本项目使用的是以CP2102芯片为核心的适配器，配合PC端的驱动软件。激光雷达装载于AGV的尾部，其用于定位的“激光雷达坐标”与“AGV运动坐标”须相重合或平行。



**图3.8 改进版本的Rao-Blackwellization粒子滤波算法伪代码[20]**

软件方面，建图功能的实现用的是ROS系统下的Gmapping功能包。此包的核心算法是一种改进版的Rao-Blackwellization粒子滤波算法（Rao-Blackwellization Particle Filter, RBPF）。根据文献[21]记载，应用在SLAM中的RBPF算法的核心在于估计后验概率，即在给定观测值和里程计的值的情况下，估计出地图信息和机器人实际轨迹。然而地图信息的构建依赖于机器人的位置，二者相互耦合，因此需要用条件概率公式将其分开。实际操作中可以用如下公式，先估计机器人位置，再用估计的位置来构建地图：

RBPF算法的基本步骤[20]如下：

1）采样：新的粒子位置是由上一时刻的粒子位置根据提议分布计算出的；

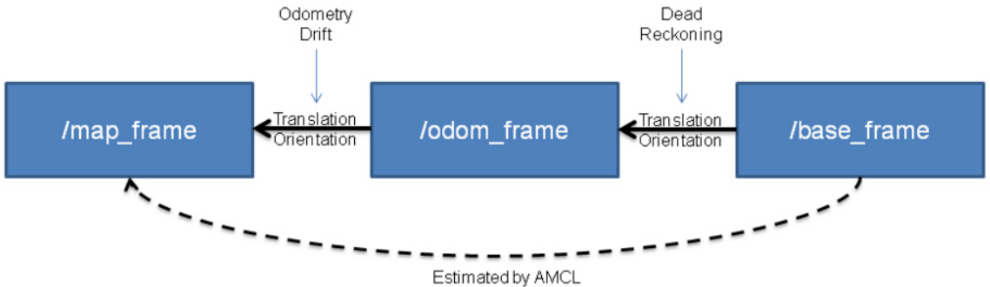
2）计算权重：作为计算权重的递推形式，用于衡量提议分布和实际分布的差异大小；

3）重采样：粒子向权重高的位置聚集；

4）估计地图：运用公式来估计地图信息。

而机器人操作系统ROS中Gmapping包实现的是改进版本的RBPF算法。改进版本主要解决了两个原始版本的问题：“找合适的提议分布”和“重采样造成的粒子枯竭”。为了解决前者，构造提议分布的时候，除了基于里程计的随机运动模型外，还使用了当前的激光雷达观测值。同时文章[20]还给出了计算权重的有效算法：

为了解决后者，算法采用了“自适应的重采样”策略，具体内容可参考文献[20]。

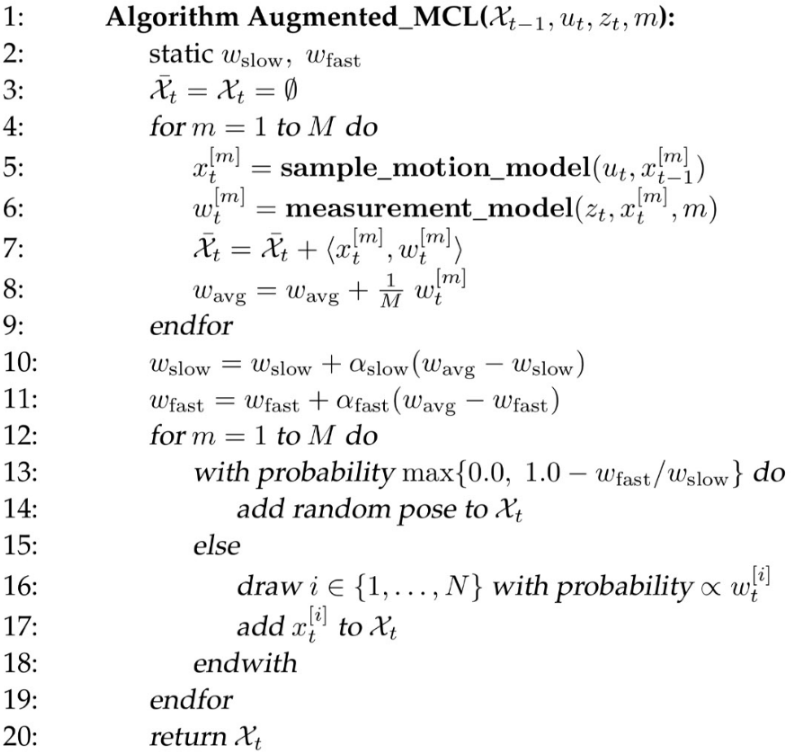


**图3.9 定位算法维护的三个坐标系的关系[15]**

软件中的定位功能的实现用的是ROS系统下的AMCL功能包。理想情况下，在已知“轮子转速”和“AGV移动速度”之间的运动学参数的基础上，仅使用里程计信息就可以确定任意时刻AGV的位置（即和之间的坐标变换关系）。然而由于有轮子相对地面的滑动、里程计信息的误差、模型简化等引入噪声的途径，仅s仅使用里程计信息就会造成误差累计导致定位偏离实际位置。而使用基于激光雷达信号的定位算法就是为了纠正这一误差（定位算法计算和之间的坐标变换关系）。

AMCL是一种针对移动机器人的二维随机定位算法包，它本质上是实现了一种自适应的蒙特卡洛定位算法。其主要算法的实施步骤如下图所示(本节末)。

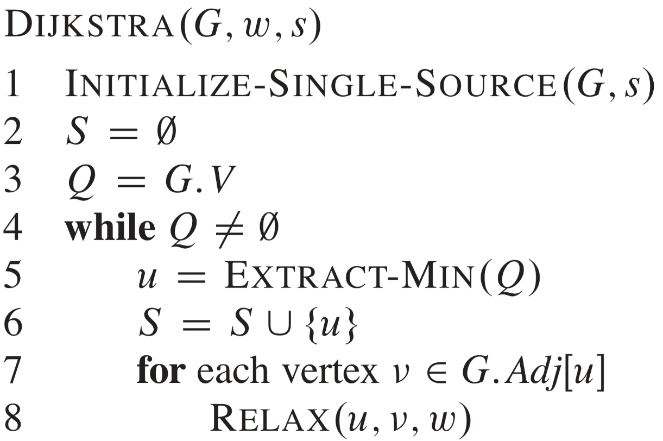
其中函数实现的是根据时刻机器人的位置和当前输入推断时刻的机器人位置（其中m为当前粒子的序号1~M）。函数的作用是根据当前雷达传感器的测量值和前一步推断出的来赋予“当前粒子点”一个权重。随后的循环进行的是“重采样”步骤，目的是让采样点例子向概率较大的点处聚集。事实上蒙特卡洛定位法的各种变形大多是针对重采样步骤的改进，例如随机粒子蒙特卡洛定位、Kullback-Leibler散度粒子蒙特卡洛定位等。当然，“运动估计”和“传感器信号识别和地图建立”两个模块也有许多变种形式。前者主要有“速度运动模型”与“里程计运动模型”两大类；后者主要有“激光雷达测距波束模型”、“可能性域模型”、“相关性模型”、“基于特征的模型”四大类。这些函数不同的实现方式的相互组合，就构成了不同的定位算法（Localization）模块。



**图3.10 实现AMCL算法的伪代码[22]**

**3.3.4路径规划(Planning)模块**

此项目中的路径规划采用move\_base开源ROS包，路径规划包含两个模块：全局路径规划（global planner）模块负责从起点到终点的总的路径规划，而局部路径规划（local planner）模块负责根据传感器探测到的AGV周围障碍物进行避障行进。



**图3.11 Dijkstra算法伪代码[23]**

全局路径规划使用navfn包实现，此包主要是实现了图论中的最短路径Dijkstra算法。此算法是解决单源最短路径的问题的经典算法。算法的流程见上图所示首先初始化需要搜索的图（此项目中是栅格状地图）。算法需要维护两个集合和一个优先队列（最小堆）：“已知距离的点集合S”、“未知距离的点集合Q”、“相对已知点距离的优先队列P”。算法流程就是从Q中选取“P最小”的点加入S集合，并根据放入的点更新（Relax）优先队列的值，直至Q成为空集合为止。Dijkstra算法从原理上讲属于贪心算法的一种，因为每一步都是选择局部的最优解，最终合成全局最优解（最短路径）。

局部路径规划使用base\_local\_planner包实现，主要算法是动态窗口法（Dynamic Window Approach, DWA）。算法思路和流程是：在二维速度空间中采样，由于当前速度、配置文件和电机性能的限制，所以速度采样范围是一个窗口。进而TrajectorySampleGenerator函数根据选定的机器人运动学模型（全向移动机器人模型与非全向移动模型）产生一系列轨迹线路。再由TrajectoryCostFunction对轨迹进行遍历操作并计算代价，归一化求和。最后找到最优轨迹作为AGV的局部规划运动的路径。其中在计算轨迹代价步骤中，用到了多种不同类型的评价函数，如震荡代价、与障碍物之间的距离代价、是否靠近全局路径、是否朝向最终目标等，用以对轨迹进行多维度评价。

**3.4 实验与测试**

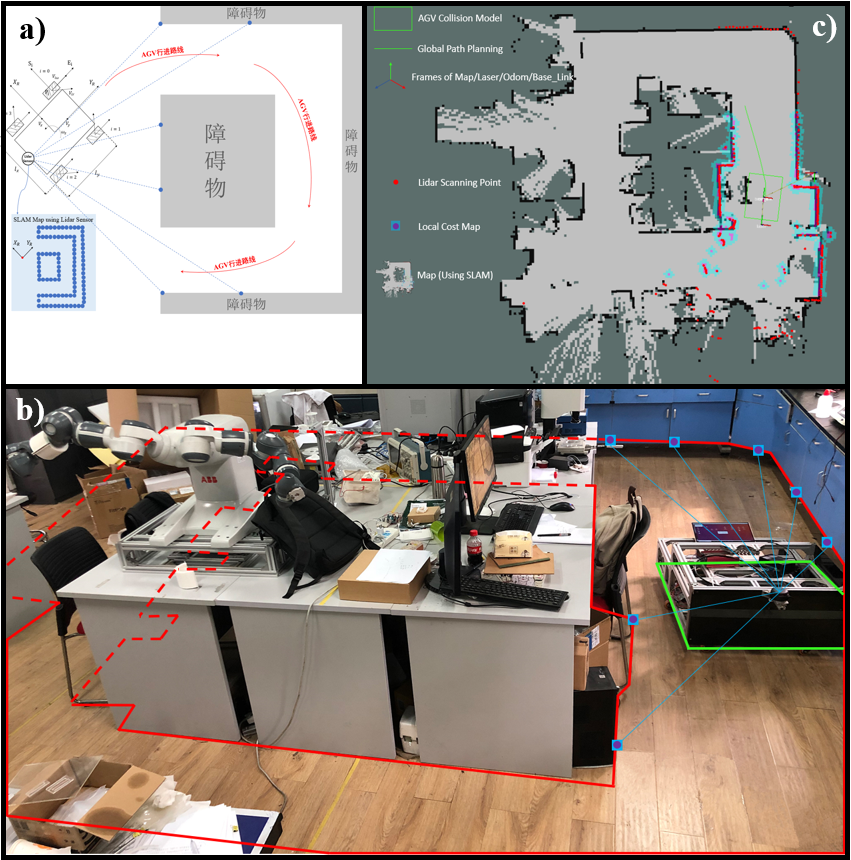
AGV自动导航功能的测试工作分为三个阶段：

（1）首先需要测试“上位机ROS系统控制AGV主控板”功能。确保上位机指令帧的编码，以及下位机的解码操作正确，指令能转化为AGV的实际运动速度。

（2）其次需要测试“实时建图与定位（SLAM）”功能。使用AGV控制指令使其沿着通道行进一圈。行进途中开启建图与定位功能包（slam\_gmapping）建立如下图所示的二维点阵地图。并将地图以pgm的格式保存在ROS系统数据库中。（3）最后测试定位与导航功能。开启定位功能包（AMCL）以及导航功能包（move\_base），验证AGV行进过程中能识别自身所处的地图位置，以及能正确规划从起点到终点的“全局路线”以及能正确规划避开障碍物的“局部路线”。

以上三个阶段需要按顺序进行，因为后一阶段的功能是在前一阶段的基础上实现的，调试环节相互耦合。例如：实时建图与定位功能需要操作者控制AGV绕着整个实验空间行进，因此需要用到“ROS操控AGV”的功能。而定位与导航功能则需要建立在SLAM步骤所创建的地图“能正确反映周围环境”的基础之上。

下图a)展示了整个实验环境的俯视示意图。整个行进过程要走两遍，第一遍由操作者通过ROS系统向下位机发送指令，操控AGV沿着通道行进，并通过SLAM功能建立地图并保存。第二遍将AGV停放于起始位置，开启自动导航模块包，由使用者指定终点位置，ROS软件系统进行路径规划、导航控制。最终实现AGV的自动导航功能。



**图3.13 a)自动引导运输车的实验设；b)实验实施情；c)实验结果**

在现有程序框架下，系统基本可以实现建图（SLAM）、定位、路径规划和运动控制等设计的功能。如上图c)所示，通过AGV车载激光雷达可以建立较高精度的地图，并且实现了实时定位功能；同时ROS系统还在起始点到目标点之间进行了路径规划，并通过串口向下位机发送了相应的运动指令。但是由于硬件条件的限制，在功能的精度上离实际应用在家庭看护方面还有一定的差距。

建立地图（SLAM）过程中，由于ROS系统无法获得“里程计信息”，因此采用“速度指令”进行替代，所以里程计坐标odom与实际机器人位置有较大差异，导致通过ROS系统建立的地图精度无法达到要求。目前的解决办法是采用AGV自带的SLAMWARE高精度定位建图硬件模块，通过读取电机码盘信号和激光雷达信号来进行实时定位与建图。而如果需要将建图功能同ROS其他模块结合起来，根本的解决办法是更改AGV主控板程序，使ROS系统获得伺服电机码盘信号从而计算出更准确的里程计坐标。同样由于缺少精确的里程计信息，实时定位的精度也不够高。

**3.5 本章小结**

本章主要阐述了AGV系统的研究和开发工作，设计三个方面的内容：机械结构、电气结构、机器人操作系统ROS的基础原理和各个模块的功能。

（1）车辆机械结构方面：根据文献记载的方法和AGV实际机械参数，推导出了此项目使用的四麦克纳姆轮AGV的运动学雅可比矩阵。继而根据此运动学变换可以获得AGV整体“前进、后退、左移、右移、左旋、右旋”六个方向的速度下，所对应的四个麦克纳姆轮的旋转角速度。是对AGV主控制器进行有关“电机控制”的设计的基础。

（2）车辆电路结构方面：描述了本项目中使用的FD123-CA-000驱动器接口以及CANopen等通信协议。继而对SMC60S-0020-30AAK直流伺服电机电流环、速度环、位置环的三环控制策略进行了理论上的分析，并结合驱动器原理和参数，描述了AGV上的直流伺服电机的具体调试方法。

（3）机器人操作系统ROS的基础内容：简要描述了机器人操作系统节点（Node）的含义以及自定义消息（Message）通过话题（Topic）的通信机制。进而对机器人操作系统ROS上的编译工具catkin\_make以及编译流程进行了讲解。

（4）ROS与AGV控制器之间的通信：使用的是针对AGV主控制器专门设计编写的“ROS系统串行通信模块”。设计过程参考了计算机网络OSI模型分层设计的思想。其中主要用到了OSI七层模型中的物理层（连接线缆类型）、数据链路层（CRC循环冗余校验码算法）以及应用层（AGV控制帧的构成）。

（5）建图与定位（Mapping and Localization）模块：硬件方面采用思岚公司的RPLIDAR A2型号激光雷达产品。软件方面，建图功能的实现用的是ROS系统下的Gmapping功能包。此包的核心算法是一种改进版的Rao-Blackwellization粒子滤波算法（Rao-Blackwellization Particle Filter, RBPF），算法的执行流程和原理已在对应小节详细描述。定位功能的实现用的是ROS系统下的AMCL功能包。AMCL是一种针对移动机器人的二维随机定位算法包，它本质上是实现了一种自适应的蒙特卡洛定位算法。算法的执行流程和原理已在对应小节详细描述。引入这种算法的目的是弥补轮子相对地面的滑动、里程计信息的误差、模型简化等引入的误差。

（6）路径规划模块（Planning）：此项目中的路径规划采用move\_base开源ROS包。路径规划包含两个模块，其中全局路径规划（global planner）模块负责从起点到终点的总的路径规划，使用navfn包实现，本章具体介绍了全局路径规划使用的Dijkstra算法的实现过程。而局部路径规划（local planner）模块负责根据传感器探测到的AGV周围障碍物进行避障行进，使用base\_local\_planner包实现，本章介绍了局部路径规划使用的动态窗口法（Dynamic Window Approach, DWA）的算法实现流程。

# **第4章 总结与展望**

**4.1 工作总结**

本文的工作集中于“人机交互系统”中的“可穿戴设备”和“智能机器人”两方面。开发了一款能捕捉人体动作的衣服，此人机交互设备可以作为家庭护理机器人人机协作系统的输入设备。同时此项目对一款自动引导运输车（AGV）进行了开发，使其拥有了智能移动装置的功能。

其中可穿戴设备方面，本项目设计并制作了一款智能运动监测衣及附属设备。其主要功能有：1）监测人体上肢动作；2）血氧含量、心率；3）语音、触觉等多人机交互通道；4）与Android系统应用程序交流数据，并进行可视化显示。工作内容涵盖机械、电路、软件方面的设计和开发。智能机器人方面，本项目对一款自动引导运输车进行了软件和控制层面的开发，实现了1）基于机器人操作系统ROS的上下位机通信（控制指令的传输）；2）基于ROS系统的AGV建图、定位、路径规划功能。

**4.2 研究展望**

本项目实现的功能较多，内容涵盖的范围非常广，因此部分功能未进行深入的开发和优化。未来第二代产品可以考虑针对以下内容进行优化：

（1）本项目开发的“第一代智能动作捕捉衣”在穿戴起始阶段需要进行手工校准，未来可以考虑加入自动校准环节，以加快部署速度，优化穿戴体验。

（2）动作捕捉对象仅限于人体上肢肩关节、肘关节的弯曲动作，未来可以增加传感器部署的关节数量及增加能够捕捉的动作的自由度。

（3）智能运动衣的走线比较复杂。即“有线连接、并行接口的方式”的可扩展性比较差。未来可以考虑用“串行总线或无线传输”的方式对振动马达和传感器进行连接。

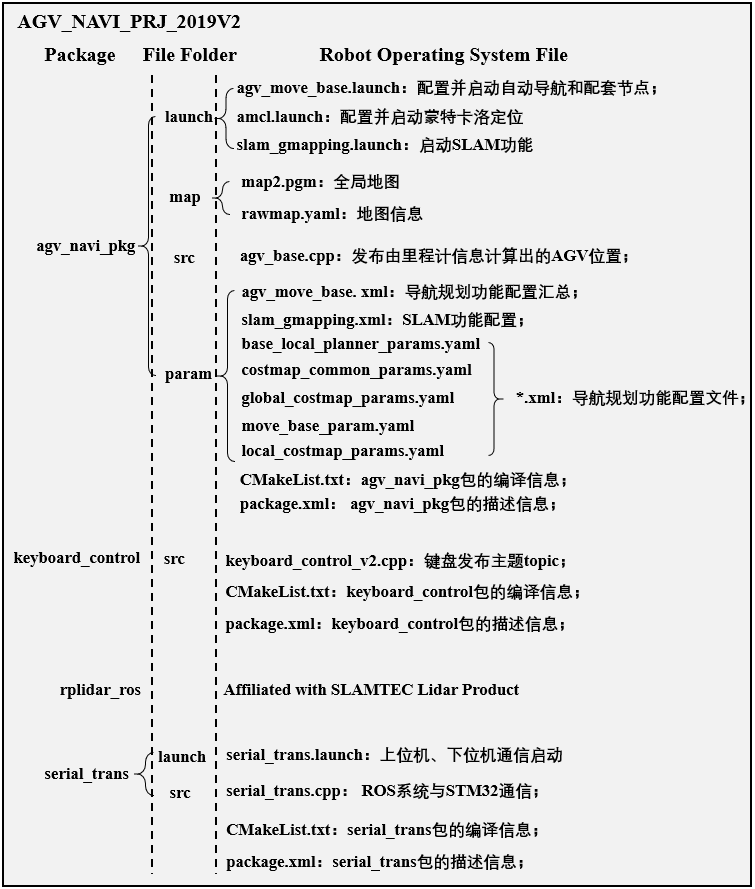
（4）AGV建图、定位、导航功能用到的ROS包来自于其他开源项目，本项目未对此进行优化。未来可以针对Slam\_gmapping包和AMCL包中的核心算法进行改进。

# **参考文献**

1. 张含阳. 人机协作：下一代机器人的必然属性[J]. 机器人产业, 2006.
2. World Health Organization. Multisectoral action for a life course approach to healthy ageing: Draft global strategy and plan of action on ageing and health [C]. 69th World Health Assembly, provisional agenda item 13.4, April 2016.
3. 2006年第二次全国残疾人抽样调查主要数据公报(第一号)[DB/OL]. https://www.cdpf.org.cn, 2006(12).
4. Sicilisno B, Khatib O, et al. Springer Handbook of Robotics [M]. Springer, 2017:1135-1346.
5. Parasuraman R, Sheridan T B, Wickens C D. A model for types and levels of human interaction with automation [J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2000, 30: 286-297.
6. Vorobyov M, Bubovich A, Galkin I. Initial in-the-field evaluation of wheelchair’s machine-to-human haptic feedback with 3D printed vibration actuator [C]. IEEE International Conference on Industrial Cyber-Physical Systems, 2018.
7. Liu H, Wang L. Gesture recognition for human-robot collaboration: A review [J]. International Journal of Industrial Ergonomics, 2018.
8. DiPietro L, Sabatini A M, DARIO P. A survey of glove-based systems and their applications [J]. IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2008, 38(4).
9. Sturman D J, Zeltzer D. A design method for “whole-hand” human- computer interaction [J]. ACM Transaction on Information and Systems，1993，11(3): 219-238.
10. Hernandez-Rebollar J L, Kyriakopoulos N K, Lindeman R W. The Accele- Glove: A whole-hand input device for virtual reality [C]. In Process of SIGGRAPH，2002: 259.
11. Atalay A, Sanchez V, Atalay O, Vogt D M, et al. Batch fabrication of customizable silicone-textile composite capacitive strain sensors for human motion tracking [J]. Advanced Materials Technologies, 2017(1700136).
12. Khan Y, Ostfeld A E, Lochner C M, et al. Monitoring of Vital Signs with Flexible and Wearable Medical Devices [J]. Advanced Materials, 2016, 28(22): 4373-4395.
13. 张蕾蕾, 朱辰, 郑敏, 等. 耳部佩戴型反射式无线血氧监测仪的研制[J]. 生物医学工程进展, 2018，39(3)：130-134.
14. ICRoute. LD3320数据手册[K]. 2010.
15. ROS WiKi [DB/OL]. http://www.ros.org/about-ros/，2018.
16. Fairchild C, Harman T L. ROS Robotics By Example [M]. Packt Publishing, 2016.
17. Cadena C, Carlone L, Carrillo H, et al. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust- Perception Age [C]. IEEE Transactions on Robotics, 2016, 32(6): 1309-1332.
18. Hamid Taheri, Bing Qiao, Nurallah Gaeminezhad, Kinematic Model of a Four Mecanum Wheeled Mobile Robot [J]. Intl. Journal of Computer Application, 2015, 113: 0975-8887.
19. Kinco. FD1X3伺服驱动器使用手册[K]. 2017.
20. Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters [M]. IEEE Transactions on Robotics, 2007, 23(1).
21. K. Murphy. Bayesian Map Learning in Dynamic Environments [C]. Conference on Neural Information Processing Systems (NIPS), Denver, 1999.
22. Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic Robotics [M]. Cambridge: The MIT Press, 2005.
23. Thomas H. Cormen, Charles E. Leiserson, et al. Introduction to Algorithm, 3rd Edition [M]. Cambridge: The MIT Press, 2009.

# **附件一：程序**

**第一部分：Robot Operating System程序**



（具体文件以电子版附件的形式上传）

**第二部分：智能动作捕捉衣程序**



（具体文件以电子版附件的形式上传）

ArduinoMega程序如下：

#include *<Wire.h>*

#include *"AngResistor.h"*

#include *"Broadcast\_Content.h"*

#include *"heartRate.h"*

#include *"MAX30105.h"*

MAX30105 particleSensor;

**const** byte RATE\_SIZE = 4; *//Increase this for more averaging. 4 is good.*

byte rates[RATE\_SIZE]; *//Array of heart rates*

byte rateSpot = 0;

long lastBeat = 0; *//Time at which the last beat occurred*

float beatsPerMinute;

int beatAvg;

*//计算血氧、心率指标的函数*

void CalculateIrBeat(long &irValue, float &beatsPerMinute, int &beatAvg)

{

irValue = particleSensor.getIR();

**if** (checkForBeat(irValue) == true)

{

*//We sensed a beat!*

long delta = millis() - lastBeat;

lastBeat = millis();

beatsPerMinute = 60 / (delta / 1000.0);

**if** (beatsPerMinute < 255 && beatsPerMinute > 20)

{

rates[rateSpot++] = (byte)beatsPerMinute; *//Store this reading in the array*

rateSpot %= RATE\_SIZE; *//Wrap variable*

*//Take average of readings*

beatAvg = 0;

**for** (byte x = 0; x < RATE\_SIZE; x++)

beatAvg += rates[x];

beatAvg /= RATE\_SIZE;

}

}

}

*//-----------------set up for time counting-----------------*

unsigned long InitialTime = 0;

unsigned long NowTime = 0;

*//-------------------set up for flags-----------------------*

int flag\_choose = 0; *//mode choose flag: state2*

int flag\_mute = 0; *//静音标记*

int flag\_signal = 0; *//voice recognition flag: state0 ~ state4*

int flag\_dash = 0; *//approaching to the end*

int actionStage = 3; *//运动姿势阶段*

int flag\_qualify = 0;*//姿势合格标记*

int ang\_left\_1\_std = 0; *//左手-上肩-标准标记*

int ang\_left\_2\_std = 0; *//左手-肘部-标准标记*

int ang\_left\_3\_std = 0; *//左手-下肩-标准标记*

int ang\_right\_1\_std = 0; *//右手-上肩-标准标记*

int ang\_right\_2\_std = 0; *//右手-肘部-标准标记*

int ang\_right\_3\_std = 0; *//右手-下肩-标准标记*

int flag\_left\_1\_vib = 0; *//flag-左手-上肩-震动次数标记*

int flag\_left\_2\_vib = 0; *//flag-左手-肘部-震动次数标记*

int flag\_left\_3\_vib = 0; *//flag-左手-下肩-震动次数标记*

int flag\_right\_1\_vib = 0; *//flag-左手-上肩-震动次数标记*

int flag\_right\_2\_vib = 0; *//flag-左手-肘部-震动次数标记*

int flag\_right\_3\_vib = 0; *//flag-左手-下肩-震动次数标记*

*//-----------------set up for bend sensor detection-----------------*

int analogPin1 = A0;*//A0~A7接收BendSensor模拟转数字信号*

int analogPin2 = A1;

int analogPin3 = A2;

int analogPin4 = A3;

int analogPin5 = A4;

int analogPin6 = A5;

int analogPin7 = A6;

float bend\_val1 = 0;*//bend\_val1~7接收弯曲传感器信号*

float bend\_val2 = 0;

float bend\_val3 = 0;

float bend\_val4 = 0;

float bend\_val5 = 0;

float bend\_val6 = 0;

float bend\_val7 = 0;

**const** float lower\_left\_1\_1 = 1.376; *//下限-左手-上肩-Stage1*

**const** float upper\_left\_1\_1 = 2.064; *//上限-左手-上肩-Stage1*

**const** float lower\_left\_2\_1 = 0.52; *//下限-左手-肘部-Stage1*

**const** float upper\_left\_2\_1 = 0.78; *//上限-左手-肘部-Stage1*

**const** float lower\_left\_3\_1 = 1.16; *//下限-左手-下肩-Stage1*

**const** float upper\_left\_3\_1 = 1.74; *//上限-左手-下肩-Stage1*

**const** float lower\_right\_1\_1 = 0.872; *//下限-右手-下肩-Stage1*

**const** float upper\_right\_1\_1 = 1.308; *//上限-右手-下肩-Stage1*

**const** float lower\_right\_2\_1 = 0.736; *//下限-右手-肘部-Stage1*

**const** float upper\_right\_2\_1 = 1.104; *//上限-右手-肘部-Stage1*

**const** float lower\_right\_3\_1 = 1.36; *//下限-右手-上肩-Stage1*

**const** float upper\_right\_3\_1 = 2.04; *//上限-右手-上肩-Stage1*

**const** float lower\_left\_1\_3 = 1.224; *//下限-左手-上肩-Stage3*

**const** float upper\_left\_1\_3 = 1.836; *//上限-左手-上肩-Stage3*

**const** float lower\_left\_2\_3 = 0.736; *//下限-左手-肘部-Stage3*

**const** float upper\_left\_2\_3 = 1.104; *//上限-左手-肘部-Stage3*

**const** float lower\_left\_3\_3 = 0.808; *//下限-左手-下肩-Stage3*

**const** float upper\_left\_3\_3 = 1.212; *//上限-左手-下肩-Stage3*

**const** float lower\_right\_1\_3 = 0.84; *//下限-右手-下肩-Stage3*

**const** float upper\_right\_1\_3 = 1.26; *//上限-右手-下肩-Stage3*

**const** float lower\_right\_2\_3 = 0.912; *//下限-右手-肘部-Stage3*

**const** float upper\_right\_2\_3 = 1.368; *//上限-右手-肘部-Stage3*

**const** float lower\_right\_3\_3 = 1.001; *//下限-右手-上肩-Stage3*

**const** float upper\_right\_3\_3 = 1.788; *//上限-右手-上肩-Stage3*

void setup(){

*//----------------set up for serial port--------------------*

Serial.begin(115200);

Serial1.begin(9600);

Serial2.begin(9600);*//broadcast*

Serial3.begin(9600);*//voice recognition*

*//-----------------set up for heartrate--------------------*

Serial.println("Initializing...");

**if** (!particleSensor.begin(Wire, I2C\_SPEED\_FAST)) *//Use default I2C port, 400kHz speed// Initialize sensor*

{

Serial.println("MAX30105 was not found. Please check wiring/power. ");

**while** (1);

}

Serial.println("Place your index finger on the sensor with steady pressure.");

particleSensor.setup(); *//Configure sensor with default settings*

particleSensor.setPulseAmplitudeRed(0x0A); *//Turn Red LED to low to indicate sensor is running*

particleSensor.setPulseAmplitudeGreen(0); *//Turn off Green LED*

*//----------------set up for vibrator pin------------------*

pinMode(22,OUTPUT);*//左手-上肩-1//Digital pin 22~35: Vibrator 14个*

pinMode(23,OUTPUT);*//左手-上肩-2*

pinMode(24,OUTPUT);*//左手-肘部-1*

pinMode(25,OUTPUT);*//左手-肘部-2*

pinMode(26,OUTPUT);*//左手-下肩-1*

pinMode(27,OUTPUT);*//左手-下肩-2*

pinMode(28,OUTPUT);*//右手-上肩-1*

pinMode(29,OUTPUT);*//右手-上肩-2*

pinMode(30,OUTPUT);*//右手-肘部-1*

pinMode(31,OUTPUT);*//右手-肘部-2*

pinMode(32,OUTPUT);*//右手-下肩-1*

pinMode(33,OUTPUT);*//右手-下肩-2*

pinMode(34,OUTPUT);*//腰部1*

pinMode(35,OUTPUT);*//腰部2*

}

void loop(){

flag\_mute = 0; *//静音标记*

flag\_choose = 0; *//选择运动类型*

flag\_dash = 0; *//将要结束标记*

flag\_signal = 0; *//语音控制命令*

actionStage = 3; *//运动姿势阶段*

flag\_qualify = 0; *//姿势合格标记*

flag\_left\_1\_vib = 0; *//flag-左手-上肩-震动次数标记*

flag\_left\_2\_vib = 0; *//flag-左手-肘部-震动次数标记*

flag\_left\_3\_vib = 0; *//flag-左手-下肩-震动次数标记*

flag\_right\_1\_vib = 0; *//flag-左手-上肩-震动次数标记*

flag\_right\_2\_vib = 0; *//flag-左手-肘部-震动次数标记*

flag\_right\_3\_vib = 0; *//flag-左手-下肩-震动次数标记*

ang\_left\_1\_std = 0; *//左手-上肩-标准标记*

ang\_left\_2\_std = 0; *//左手-肘部-标准标记*

ang\_left\_3\_std = 0; *//左手-下肩-标准标记*

ang\_right\_1\_std = 0; *//右手-上肩-标准标记*

ang\_right\_2\_std = 0; *//右手-肘部-标准标记*

ang\_right\_3\_std = 0; *//右手-下肩-标准标记*

InitialTime = millis();

*//-----------------------State0: Standby--------------------------//*

**while**(1)

{

Serial.println("State0:Standby");

**if**(Serial3.available())

{

flag\_signal = Serial3.read();

}

**if**(flag\_signal == 1)*//站姿肩部运动*

{

**break**;

}

}

*//-----------------------State1: Enter--------------------------//*

*// while(1)*

*// {*

*// Serial.println("State1:Enter");*

*// Serial1.write(0xFF);*

*// Serial1.write(0xFF);*

*// Serial1.write(0x01);//开始态*

*//*

*// speech\_state1();//播报：开始运动*

*// delay(1500); //延迟1500ms*

*// break;*

*// }*

*//-----------------------State2: Choose--------------------------//*

**while**(1)

{

Serial.println("State2:Choose");

Serial1.write(0xFF);

Serial1.write(0xFF);

Serial1.write(0x02);*//选择态*

**if**(flag\_choose == 0)*//只播报一次：站姿肩部深层激活，现在开始*

{

flag\_choose = 1;

speech\_state2\_2();

}

delay(4000);

**break**;

*// if(Serial3.available())*

*// {*

*// flag\_signal = Serial3.read();*

*// }*

*// if(flag\_signal == 2)//站姿肩部运动*

*// {*

*// Serial.println(flag\_signal);*

*// break;*

*// }*

}

*//-----------------------State3: Motion--------------------------//*

int times = 0;

**while**(1)

{

*//long StartTime = millis();*

times++;

**if**(times==15 && flag\_mute==0)

{

Serial.print("Persist!!!!!!!!!!!!!!!!!!");

speech\_persist();*//播报：再坚持一下*

delay(100);

}

Serial.println(times);

Serial.println("State3:Motion");

Serial.print("flag\_mute:");

Serial.println(flag\_mute);

*//1.传感器信息采集（数模转换）部分*

bend\_val1=analogRead(analogPin1)\*5.0/1023;

bend\_val2=analogRead(analogPin2)\*5.0/1023;

bend\_val3=analogRead(analogPin3)\*5.0/1023;

bend\_val4=analogRead(analogPin4)\*5.0/1023;

bend\_val5=analogRead(analogPin5)\*5.0/1023;

bend\_val6=analogRead(analogPin6)\*5.0/1023;

bend\_val7=analogRead(analogPin7)\*5.0/1023;

long irValue = 0;

float bearPerMinute = 0;

int beatAvg = 0;

CalculateIrBeat(irValue, beatsPerMinute, beatAvg);

*//2.蓝牙传输部分(11Byte)*

Serial1.write(0xFF);

Serial1.write(0xFF);

**if**(flag\_mute == 0)

{

Serial1.write(0x03);

}

**else** **if**(flag\_mute == 1)

{

Serial1.write(0x05);

}

Serial1.write(0x01);*//7byte弯曲数值*

Serial1.write(0x01);

Serial1.write(0x01);

Serial1.write(0x01);

Serial1.write(0x01);

Serial1.write(0x01);

Serial1.write(0x01);

Serial1.write(0x00);*//1byte弯曲角度合格*

Serial1.write(beatAvg);*//1byte心率数值*

*//3.姿势判断部分*

**if**(actionStage == 1)*//第一阶段姿势判断*

{

Serial.println("actionStage1");

**if**(bend\_val1>lower\_left\_1\_1 && bend\_val1<upper\_left\_1\_1)*//上下限-左手-上肩部-Stage1*

ang\_left\_1\_std = 1;

**else**

ang\_left\_1\_std = 0;

**if**(bend\_val2>lower\_left\_2\_1 && bend\_val2<upper\_left\_2\_1)*//上下限-左手-肘部-Stage1*

ang\_left\_2\_std = 1;

**else**

ang\_left\_2\_std = 0;

**if**(bend\_val3>lower\_left\_3\_1 && bend\_val3<upper\_left\_3\_1)*//上下限-左手-下肩部-Stage1*

ang\_left\_3\_std = 1;

**else**

ang\_left\_3\_std = 0;

**if**(bend\_val4>lower\_right\_1\_1 && bend\_val4<upper\_right\_1\_1)*//上下限-右手-下肩部-Stage1*

ang\_right\_1\_std = 1;

**else**

ang\_right\_1\_std = 0;

**if**(bend\_val5>lower\_right\_2\_1 && bend\_val5<upper\_right\_2\_1)*//上下限-右手-肘部-Stage1*

ang\_right\_2\_std = 1;

**else**

ang\_right\_2\_std = 0;

**if**(bend\_val6>lower\_right\_3\_1 && bend\_val6<upper\_right\_3\_1)*//上下限-右手-上肩部-Stage1*

ang\_right\_3\_std = 1;

**else**

ang\_right\_3\_std = 0;

**if**(ang\_left\_1\_std && ang\_left\_2\_std && ang\_left\_3\_std && ang\_right\_1\_std && ang\_right\_2\_std && ang\_right\_3\_std)

{*//若合格：进入合格提醒部分*

*//若不合格：进入不合格提醒部分*

flag\_qualify = 1; *//姿势判断合格标识：flag\_qualify为界*

ang\_left\_1\_std = 0;

ang\_left\_2\_std = 0;

ang\_left\_3\_std = 0;

ang\_right\_1\_std = 0;

ang\_right\_2\_std = 0;

ang\_right\_3\_std = 0;

}

}

**if**(actionStage == 3)*//第三阶段姿势判断（第二阶段省略）*

{

Serial.println("actionStage3");

**if**(bend\_val1>lower\_left\_1\_3 && bend\_val1<upper\_left\_1\_3)*//上下限-左手-上肩部-Stage3*

ang\_left\_1\_std = 1;

**else**

ang\_left\_1\_std = 0;

**if**(bend\_val2>lower\_left\_2\_3 && bend\_val2<upper\_left\_2\_3)*//上下限-左手-肘部-Stage1*

ang\_left\_2\_std = 1;

**else**

ang\_left\_2\_std = 0;

**if**(bend\_val3>lower\_left\_3\_3 && bend\_val3<upper\_left\_3\_3)*//上下限-左手-下肩部-Stage1*

ang\_left\_3\_std = 1;

**else**

ang\_left\_3\_std = 0;

**if**(bend\_val4>lower\_right\_1\_3 && bend\_val4<upper\_right\_1\_3)*//上下限-右手-下肩部-Stage1*

ang\_right\_1\_std = 1;

**else**

ang\_right\_1\_std = 0;

**if**(bend\_val5>lower\_right\_2\_3 && bend\_val5<upper\_right\_2\_3)*//上下限-右手-肘部-Stage1*

ang\_right\_2\_std = 1;

**else**

ang\_right\_2\_std = 0;

**if**(bend\_val6>lower\_right\_3\_3 && bend\_val6<upper\_right\_3\_3)*//上下限-右手-上肩部-Stage1*

ang\_right\_3\_std = 1;

**else**

ang\_right\_3\_std = 0;

**if**(ang\_left\_1\_std && ang\_left\_2\_std && ang\_left\_3\_std && ang\_right\_1\_std && ang\_right\_2\_std && ang\_right\_3\_std)

{ *//若合格：进入合格提醒部分*

*//若不合格：进入不合格提醒部分*

flag\_qualify = 1; *//姿势判断合格标识*

ang\_left\_1\_std = 0;

ang\_left\_2\_std = 0;

ang\_left\_3\_std = 0;

ang\_right\_1\_std = 0;

ang\_right\_2\_std = 0;

ang\_right\_3\_std = 0;

}

}

*//4.姿势提醒部分*

**if**(flag\_qualify == 1) *//如果姿势标准√*

{

flag\_qualify = 0;

}

**else** **if**(flag\_qualify == 0) *//如果姿势不标准×*

{

*//---------------左手-上肩部 不标准---------------*

**if**(ang\_left\_1\_std == 0) *//左手-上肩部 不标准*

{

**if**(flag\_left\_1\_vib==5 && flag\_mute==0)*//震动5下，播报1下*

{

speech\_state3\_1(); *//播报：左手上肩部不标准*

flag\_left\_1\_vib = 0;

delay(200);

}

**else**

{

digitalWrite(22, HIGH);*//开始震动*

digitalWrite(23, HIGH);*//开始震动*

Serial.println("left-shoulder-1-unqualified");

flag\_left\_1\_vib++;

delay(200);

digitalWrite(22, LOW);*//停止震动*

digitalWrite(23, LOW);*//停止震动*

}

}

*//---------------左手-肘部 不标准---------------*

**if**(ang\_left\_2\_std == 0) *//左手-肘部 不标准*

{

**if**(flag\_left\_2\_vib==5 && flag\_mute==0)*//震动5下，播报1下*

{

speech\_state3\_5(); *//播报：左手肘部不标准*

flag\_left\_2\_vib = 0;

delay(200);

}

**else**

{

digitalWrite(24, HIGH); *//开始震动*

digitalWrite(25, HIGH); *//开始震动*

Serial.println("left-elbow-unqualified");

flag\_left\_2\_vib++;

delay(200);

digitalWrite(24, LOW); *//停止震动*

digitalWrite(25, LOW); *//停止震动*

}

}

*//---------------左手-下肩部 不标准---------------*

**if**(ang\_left\_3\_std == 0) *//左手-下肩 不标准*

{

**if**(flag\_left\_3\_vib==5 && flag\_mute==0)*//震动5下，播报1下*

{

speech\_state3\_6(); *//播报：左手下肩部不标准*

flag\_left\_3\_vib = 0;

delay(200);

}

**else**

{

digitalWrite(26, HIGH); *//开始震动*

digitalWrite(27, HIGH); *//开始震动*

Serial.println("left-shoulder-2-unqualified");

flag\_left\_3\_vib++;

delay(200);

digitalWrite(26, LOW); *//停止震动*

digitalWrite(27, LOW); *//停止震动*

}

}

*//---------------右手-下肩部 不标准---------------*

**if**(ang\_right\_1\_std == 0)

{

**if**(flag\_right\_1\_vib==5 && flag\_mute==0)*//震动5下，播报1下*

{

speech\_state3\_7(); *//播报：右手上肩部不标准*

flag\_right\_1\_vib = 0;

delay(200);

}

**else**

{

digitalWrite(28, HIGH); *//开始震动*

digitalWrite(29, HIGH); *//开始震动*

Serial.println("right-shoulder-2-unqualified");

flag\_right\_1\_vib++;

delay(200);

digitalWrite(28, LOW); *//停止震动*

digitalWrite(29, LOW); *//停止震动*

}

}

*//---------------右手-肘部 不标准---------------*

**if**(ang\_right\_2\_std == 0)

{

**if**(flag\_right\_2\_vib==5 && flag\_mute==0)*//震动5下，播报1下*

{

speech\_state3\_8(); *//播报：右手肘部不标准*

flag\_right\_2\_vib = 0;

delay(200);

}

**else**

{

digitalWrite(30, HIGH); *//开始震动*

digitalWrite(31, HIGH); *//开始震动*

Serial.println("right-elbow-unqualified");

flag\_right\_2\_vib++;

delay(200);

digitalWrite(30, LOW); *//停止震动*

digitalWrite(31, LOW); *//停止震动*

}

}

*//---------------右手-上肩部 不标准---------------*

**if**(ang\_right\_3\_std == 0)

{

**if**(flag\_right\_3\_vib==5 && flag\_mute==0)*//震动5下，播报1下*

{

speech\_state3\_9(); *//播报：右手下肩部不标准*

flag\_right\_3\_vib = 0;

delay(200);

}

**else**

{

digitalWrite(32, HIGH); *//开始震动*

digitalWrite(33, HIGH); *//开始震动*

Serial.println("right-shoulder-1-unqualified");

flag\_right\_3\_vib++;

delay(200);

digitalWrite(32, LOW); *//停止震动*

digitalWrite(33, LOW); *//停止震动*

}

}

}

*//5.状态转移部分*

flag\_signal = Serial3.read();

**if**(flag\_signal == 2)*//静音*

{

flag\_mute = 1;

}

**else** **if**(flag\_signal == 3)*//发音*

{

flag\_mute = 0;

}

**else** **if**(flag\_signal == 4)*//暂停*

{

**while**(1)

{

Serial1.write(0xFF);

Serial1.write(0xFF);

Serial1.write(0x06);

delay(500);

flag\_signal = Serial3.read();

**if**(flag\_signal == 5)*//启动*

**break**;

}

}

**else** **if**(flag\_signal == 6 || times == 20)*//结束运动*

**break**;

delay(1680);

**if**(actionStage == 1)

{

actionStage = 3;

}

**else** **if** (actionStage == 3)

{

actionStage = 1;

}

}

*//-----------------------State4: End----------------------------//*

**while**(1)

{

Serial.println("State4:End");

Serial1.write(0xFF);

Serial1.write(0xFF);

Serial1.write(0x04);

speech\_state4();

delay(3000);

**break**;

}

}

# **附件二：毕业设计成果**

EI会议文章收录：

Pan Shimin, Lv Honghao, Duan Hong, Pang Gaoyang, Yi Kang, and Yang Geng\*, "A Sensor Glove for the Interaction with a Nursing-Care Assistive Robot", in the 2019 IEEE International Conference on Industrial Cyber-Physical Systems (ICPS 2019), Taipei, 6-9 May. 2019(Accepted).

**本科生毕业论文（设计）任务书**

**一、题目：基于可穿戴式设备的人-机器人协作研究**

**二、指导教师对毕业论文（设计）的进度安排及任务要求：**

**2018.11-2018.12 撰写开题报告, 文献综述与外文翻译**

**2019.1-2019.2 学生文献综述与开题报告修改与上传**

**2019.1-2019.3 开展实验研究，收集整理数据，撰写毕业论文初稿，中期检查**

**2019.4-2019.6 进一步开展实验研究，收集整理数据，修改毕业论文，论文上传至教务系统**

**起讫日期 2018 年 11 月 25 日至 2019 年 6 月 10日**

**指导教师**（**签名） 职称**

**三、系或研究所审核意见:**

**负责人**（**签名）**

**2018年 11 月 30 日**

**毕业论文（设计）的进度安排**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 毕业论文（设计）  各阶段工作内容 | 工作进度周次安排 | | | | | | | | | | | | | | | | | | | | | | | |
| 秋冬学期 | | | | | | | | 春夏学期 | | | | | | | | | | | | | | | |
| 11 | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 91 | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 99 | 110 | 111 | 112 | 113 | 114 | 115 | 116 |
| 1.文献阅读、外文文献翻译 | — | — | — |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.完成开题报告、文献综述 |  |  |  | — | — | — | — |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.完成智能运动捕捉衣的需求分析和设计 |  |  |  |  |  |  |  | — |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.智能衣电路、软件部分的设计及制作 |  |  |  |  |  |  |  | — | — | — | — | — | — | — | — |  |  |  |  |  |  |  |  |  |
| 5.智能衣机械部分的设计及制作 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | — | — |  |  |  |  |  |  |  |  |
| 6.穿戴并调试智能衣的相应功能 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | — | — | — | — |  |  |  |  |  |
| 7.需求分析及ROS系统模块功能设计 |  |  |  |  |  |  | — | — |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8.ROS系统编程实现 |  |  |  |  |  |  |  |  | — | — | — | — | — | — | — |  |  |  |  |  |  |  |  |  |
| 9.上机调试AGV导航功能 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | — | — | — | — |  |  |  |  |  |
| 10.完成毕业论文 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | — | — | — | — | — |  |  |  |

**指导教师签名：**

**日期：2019年6月10日**

注：①本表由指导教师填写，各专业学科毕业设计负责人审核并签名。②各阶段工作内容应包括：查阅文献、文献综述、外文文献翻译、开题报告、调研、计算机仿真、设计绘图、实验、撰写毕业论文（设计）等，其中4-9栏目由指导教师布置、填写。③工作进度周次安排由指导教师在相应周次里画横线表示，（毕设24周，周次可以写为3-4周、8-9周等）。④导师应在检查各阶段工作进度完成情况后签名，此计划可根据实际情况作调整。

**毕业论文（设计）考核**

**一、指导教师对毕业论文（设计）的评语：**

**指导教师(签名)**

**年 月 日**

**二、答辩小组对毕业论文（设计）的答辩评语及总评成绩：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **成绩**  **比例** | **文献综述**  **（满分10分）** | **开题报告**  **（满分15分）** | **外文翻译**  **（满分5分）** | **毕业论文（设计）质量及答辩**  **（满分70分）** | **总评**  **成绩**  **（满分100分）** |
| **分值** |  |  |  |  |  |

**答辩小组负责人（签名）**

**年 月 日浙江大学本科生毕业论文（设计）存档资料检查表**

（此表请装订在毕业论文（设计）末页，由指导教师在毕业论文（设计）答辩结束后填写）

|  |  |  |
| --- | --- | --- |
| **检查内容** | | **检查结果（打√、X）** |
| 毕业  设计 | 1．装订（封面白色铜版纸，A4） |  |
| 2．毕业论文（设计）诚信承诺书(签名) |  |
| 3. 任务书（题目、要求、工作进度表） |  |
| 4．中、英文摘要 |  |
| 5．目录（要标注页码） |  |
| 6．正文 |  |
| 7．附件1：图纸、程序 |  |
| 8．附件2：毕业设计期间取得的成果 |  |
| 9. 参考文献 |  |
| 10.毕业设计指导教师评语、答辩评语和成绩 |  |
| 开题  报告 | 11．装订（文献综述、开题报告、外文翻译、外文原稿） |  |
| 12．文献综述、开题报告的成绩、评语 |  |
| 其它 | 13．毕业论文（设计）专家评阅意见（1份） |  |
| 14．毕业论文（设计）答辩记录表（1份） |  |
| 15. 毕业论文（设计）查重报告 |  |

注：“毕业论文（设计）专家评阅意见”、“毕业论文（设计）答辩记录表”、“毕业论文（设计）查重报告”检查后装入“毕业设计档案袋”，直接归档，不装订。

检查人（签名）： 检查日期：2019年6月10日