

摘 要

在机器人的研究领域，如何有效地提高机器人控制系统的控制性能始终是研究学者十分关注的一个重要内容。在分析了工业机器人的发展历程和机器人控制系统的研究现状后，本论文的主要目标是建立一个新型全数字的基于 ATmega128 和 FPGA 的机器人位置伺服控制系统的软、硬件平台，实现对五自由度教学机器人的精确控制。

本论文分析了所研究的五自由度教学机器人的本体结构。在明确了实现机器人精确位置伺服控制的控制原理后，在分析对机器人控制系统的诸多可行性方案的基础上，确定采用了两级 CPU 控制体系结构：第一级 CPU 为上位计算机，它实现对机器人的系统管理、协调控制以及完成机器人实时轨迹规划等控制算法的运算；第二级 CPU 为高性能的 ATmega128 处理器，辅之以具有高速并行处理能力的 FPGA 芯片，实现了对机器人多个关节的高速并行驱动；采用串口实现上位计算机与下位控制器之间的数据通信，这样既保证了两者之间连接方便，又有效的提高了控制系统的通信速度和可靠性。

机器人系统的软件设计包括两个部分：一是采用 VC++ 实现的上位监控软件系统，它主要负责机器人实时轨迹规划等控制算法的运算，同时完成用户与机器人系统之间的信息交互；二是采用实时操作系统 uC/OS II 实现的下位机控制程序，它主要负责接收上位监控系统的命令，实现对机器人的实时驱动，同时还能够实时的向上位监控系统机器人反馈的当前状态信息。

研究开发出的五自由度教学机器人控制器具有控制实时性好、定位精度高、运行稳定可靠的特点，它允许用户通过上位控制计算机实现对机器人的各种设定作业的控制。

关键词：教学机器人；多自由度伺服控制器；实时操作系统；ATmega128

Abstract

In the field of robot research, it is an important content that effectively enhances the robot control system's performance. This paper analyzes the development of industry robot and the current research situation of robot controller. The essentially targets of this article is established a new entirely digital robot position servo-control system based on the ATmega128 and FPGA for the five multi-joints robot.

This paper analyzes the body structure of the five multi-joints robot. After exact location servo-control theory of the robot is clearly realized, it studies and compares many feasible robot control system plans. Finally it decides to adopt the two-stage CPU control system structure: On the first stage is the personal computer. It realizes the administration, the co-control of the robot system as well as accomplishes the robot's real-time path plan by using advanced control algorithm. The ATmega128 with high performance is on the second stage. It is assisted by the FPGA chip with the ability of high speed and parallel processing. And it makes the robot's five joints rotate parallel with high speed. In addition, the PC and the ATmega128 communicates with UART port. The UART communication mode not only guarantees the communication's convenience, but also effectively enhances the control system's real-time and reliability.

Robot control system software design including two parts: The first part uses the VC++ to realize the robot control algorithm and the interaction between the user and the robot system in real-time. The second part is based on uC/OS II embedded system. It realizes the function of driving the robot motor by getting commands from PC, and the function of feeding back robot's real-time location and status information.

According to the above design, it works out the robot control system and finds it has a very good real-time property, high position precision, stability and reliability. It allows the user to control the robot by operating on the PC.

Keywords: teach-robot; numerous DC servomotor control; RTOS; ATmega128

哈尔滨工程大学 学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献的引用已在文中指出，并与参考文献相对应。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者（签字）：孙卓君

日期：2007年1月24日

第 1 章 绪论

1.1 本文选题目的及意义

随着现代科学技术的迅速发展,尤其是进入 90 年代以来,机器人技术的进步与其在各个领域的广泛应用,引起了各国专家学者的普遍关注。许多发达国家均把机器人技术的开发、研究列入国家高新技术发展计划。世界各国普遍在高等院校为大学本科生及研究生开设出介绍机器人技术的有关课程。

为了培养机器人开发、设计、生产、维护方面的人才,我国很多高校也为本科生和研究生开设了机器人学课程。笔者所开发的教学机器人,以示教再现控制为主,可以很好的说明工业机器人的工作原理和工作过程。将其作为《机器人学》、《机器人技术》及《机电一体化系统设计》课程及机械电子工程专业机电结合的综合教学实验设备,不仅可以使学生在轻松愉快的氛围中充分理解相关课程的专业知识,而且可以激发学生的专业学习兴趣,使他们树立系统工程概念,培养其独立开展科学研究的能力。因此,本教学机器人的研制成功,对机电一体化专业教学有着十分重要的意义。

本课题的目的是设计一个具有示教再现功能的五自由度的教学机器人,通过控制三个位置自由度,两个姿态自由度及手爪自由度,使其能在作业空间内运动,向学生生动地演示机器人的基本原理,培养其对机器人的兴趣。

随着我国高等教育的进一步发展,尤其对机器人技术方面人才的需求进一步扩大,教学机器人作为一种直观的教学方式,将抽象的机器人学,控制理论和机电一体化设计等多门学科的理论知识,蕴于具体形象实际的机器人中,让学生在实践的环境下学习理论知识,从中验证书中的各个定理及知识点,使其更好的掌握专业知识,综合了解各个学科之间的内在联系,因此对机器人的教学有很重要的意义^[1]。

1.2 国内外研究现状及分析

1.2.1 工业机器人的发展情况

工业机器人的发展始于 20 世纪年代, 1954 年美国戴沃尔发表了“通用重复型机器人”的专利论文, 第一次提出了“工业机器人”的概念。1958 年美国联合控制公司研制出第一台数控工业机器人原型。1959 年美国 UNIMATION 公司推出第一台工业机器人。美国是机器人的故乡。

60 年代随着传感技术和工业自动化的发展, 这时是工业机器人的发展初期, 机器人开始向适用化发展, 并被用于电焊和喷涂作业。70 年代随着计算机和人工智能的发展, 机器人进入适用化时代。日本虽起步较晚, 但结合国情, 面向中小企业, 采取了一系列鼓励使用机器人的措施。其机器人拥有量很快超过了美国, 一举成为“机器人王国”。

80 年代工业机器人进入发展中期, 汽车、电子等行业开始大量使用工业机器人, 推动了机器人产业的发展。工业机器人的应用满足了人们特性化的要求, 产品的批量越来越大, 品种越来越多, 而且产品的一致性也大大提高, 为商家占有了更多的市场份额, 获得了更多的市场利润。

90 年代初期, 工业机器人的生产与需求达到了一个技术成熟期, 1990 年世界上新装备工业机器人 80943 台, 1991 年装备了 76443 台, 到 1991 年底世界上已有 53 万台工业机器人工作在各条战线上^[2]。

1.2.2 国外发展现状

国外工业机器人领域发展近几年有如下几个趋势:

(1) 工业机器人性能不断提高(高速度、高精度、高可靠性、便于操作和维修), 而单机价格不断下降, 平均单机价格从 1991 年的 10.3 万美元降至 1997 年的 6.5 万美元。

(2) 机械结构向模块化、可重构化发展。例如关节模块中的伺服电机、减速机、检测系统三位一体化; 关节模块、连杆模块用重组方式构造机器人

整机；国外已有模块化装配机器人产品问世。

(3) 工业机器人控制系统向基于 PC 机的开放型控制器方向发展，便于标准化、网络化；器件集成度提高，控制柜日见小巧，且采用模块化结构；大大提高了系统的可靠性、易操作性和可维修性。

(4) 机器人中的传感器作用日益重要，除采用传统的位置、速度、加速度等传感器外，装配、焊接机器人还应用了视觉、力觉等传感器，而遥控机器人则采用视觉、声觉、力觉、触觉等多传感器的融合技术来进行环境建模及决策控制；多传感器融合配置技术在产品化系统中已有成熟应用。

(5) 虚拟现实技术在机器人中的作用已从仿真、预演发展到用于过程控制，如遥控机器人操作者产生置身于远端作业环境中的感觉来操纵机器人^[3]。

(6) 当代遥控机器人系统的发展特点不是追求全自治系统，而是致力于操作者与机器人的人机交互控制，即遥控加局部自主系统构成完整的监控遥控操作系统，使智能机器人走出实验室进入实用化阶段。美国发射到火星上的“索杰纳”机器人就是这种系统成功应用的最著名实例^[4]。

(7) 机器人化机械开始兴起。从 1994 年美国开发出“虚拟轴机床”以来，这种新型装置已成为国际研究的热点之一，纷纷探索开拓其实际应用的领域^[5]。

教学机器人作为机器人教学的基础，在机器人教育方面有重要的意义，因此各国都致力于教学机器人平台的开发与研制。例如德国研制了新型的教学机器人实验平台，如图 1.1 所示。其由一个五自由度的机械臂和一个视觉系统构成。机械臂有底部回转、大臂俯仰、小臂俯仰三个位置自由度和小臂回转、手腕回转两个姿态自由度，手腕上装有电机，以控制手爪的开合。通过摄像头采集的视觉信号作为机械臂的末端位置反馈到上位机中，在上位机中进行视觉伺服的算法，然后将各个关节的位置输出给机械手的伺服控制器，从而进行末端位置的控制，这样构成了一个具有视觉反馈功能的、简单的教学机器人实验的平台^[6]。

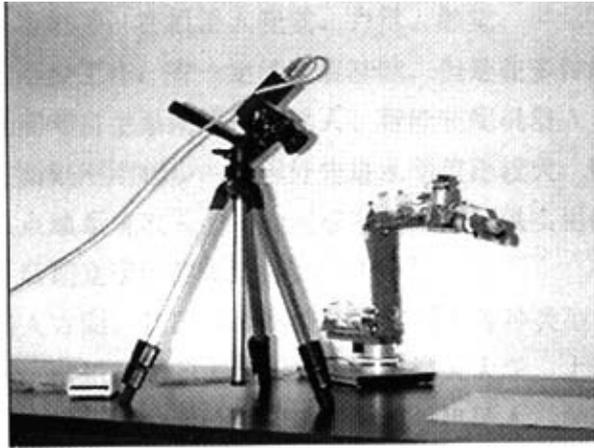


图 1.1 德国新型教学机器人实验平台

1.2.3 国内发展现状

我国的工业机器人从80年代“七五”科技攻关开始起步，在国家的支持下，通过“七五”、“八五”科技攻关，目前已基本掌握了机器人操作机的设计制造技术、控制系统硬件和软件设计技术、运动学和轨迹规划技术，生产了部分机器人关键元器件，开发出喷漆、弧焊、点焊、装配、搬运等机器人；其中有130多台套喷漆机器人在二十余家企业的近30条自动喷漆生产线（站）上获得规模应用，弧焊机器人已应用在汽车制造厂的焊装线上。

但总的来看，我国的工业机器人技术及其工程应用的水平和国外相比还有一定的距离，如：可靠性低于国外产品；机器人应用工程起步较晚，应用领域窄，生产线系统技术与国外相比有差距；在应用规模上，我国已安装的国产工业机器人约200台，约占全球已安装台数的万分之四。以上原因主要是没有形成机器人产业，当前我国的机器人生产都是应用户的要求，“一客户，一次重新设计”，品种规格多、批量小、零部件通用化程度低、供货周期长、成本高，而且质量、可靠性不稳定。因此迫切需要解决产业化前期的关键技术，对产品进行全面规划，搞好系列化、通用化、模化设计，积极推进产业化进程。我国的智能机器人和特种机器人在“863”计划的支持下，也取得了不少成果。其中最为突出的是水下机器人，6000米水下无缆机器人的成果居世界领先水平，还开发出直接遥控机器人、双臂协调控制机器人、爬壁机器

人、管道机器人等机种；在机器人视觉、力觉、触觉、声觉等基础技术的开发应用上开展了不少工作，有一定的发展基础。但是在多传感器信息融合控制技术、遥控加局部自主系统遥控机器人、智能装配机器人、机器人化机械等的开发应用方面则刚刚起步，与国外先进水平差距较大，需要在原有成绩的基础上，有重点地系统攻关，才能形成系统配套可供实用的技术和产品，有望在“十五”后期立于世界先进行列之中^[6]。

在教学机器人方面，我国各大高校分别研制了各种类型的教学机器人。主要有清华大学、哈尔滨工业大学、北京航空航天大学、上海交通大学等。

哈尔滨工业大学研制的四轴直角坐标系教学机器人，如图 1.2 所示。这个教学机器人是一种串联式的机器人，由 X、Y、Z 三个正交的自由度构成。串联式机器人是一种典型的工业机器人，在自动搬运、装配、焊接、喷涂等工业现场中有着广泛的应用，通过该系列教学机器人可使学生能够模拟工业现场的实际运行状况。其结构紧凑，工作范围大，具有高度的灵活性，是进行运动规划和编程系统设计的理想对象^[9]。

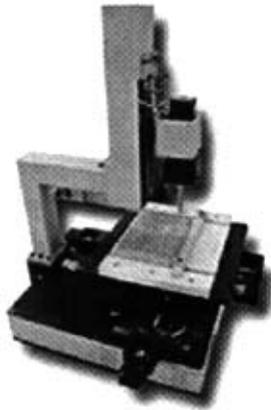


图1.2 哈尔滨工业大学研制的四轴直角坐标系教学机器人

上海交通大学研制的教学机器人如图1.3所示。其属于典型的关节式机器人，包括底部回转、大臂俯仰、小臂俯仰三个位置自由度和小臂回转、手腕俯仰、手腕回转三个姿态自由度，展示了现代工业机器人的一般特征；它能够演示现代工业机器人的典型应用；在一定程度上满足机器人理论研究的需要^[10]。



图1.3 上海交通大学的六自由度教学机器人

1.3 论文主要工作内容

本课题主要研究五自由度教学机器人样机控制系统的研制，包括：硬件电路的设计和调试、软件系统的设计和调试两大部分。论文的主要内容如下：

(1) 教学机器人的系统设计, 主要包括：教学机器人的机械结构；教学机器人的控制系统结构。

(2) 硬件控制系统的设计, 主要包括：电机驱动模块设计；基于FPGA的电机伺服模块设计；单片机控制系统硬件设计。

(3) 软件控制系统的设计, 主要包括：数字PID控制算法设计；uC/OS II实时操作系统平台的构造；与上位机通信协议的设计。

(4) 样机的研制与实验, 主要包括：硬件系统的调试；软件系统的调试；控制系统性能的调试。

第 2 章 机器人的系统方案设计

2.1 机械结构方案

本机器人系统主要作为教学研究用，其工作环境为教学实验室，要求它能够进行连续轨迹控制，同时应该具有示教再现功能。机器人的主要参数包括：坐标形式、各个杆件的长度、关节的运动范围、运动速度、运动精度和负载能力等。

由于机器人是教学实验机器人，主要用来完成一些实验、演示机器人的基本功能、完成机器人的操作方法训练，以便了解机器人的工作原理和使用方法。因此对机器人的负载能力、运动速度要求较低，希望系统的结构尽量简单、成本低，控制功能比较强。由此对研制的五自由度教学机器人提出以下几点技术要求：

- (1) 重量轻，不能超过 5 公斤，要求结构紧凑；
- (2) 机械臂末端手爪的误差不超过 5mm；
- (3) 要求手爪能抓取 1 公斤的物体。

2.1.1 坐标形式

工业机器人按照结构坐标特点可分为以下几种如图 2.1 所示。

(1) 直角坐标型三个关节运动方向互相垂直，其控制方案和数控机床相似。这一结构各轴线位移分辨率在操作空间内任一点上均为恒定，但是操作灵活性较差，体积比较大。

(2) 圆柱坐标型在水平转台上装有立柱，水平臂可沿立柱作上下运动并可在水平方向伸缩。这种结构优点是动作过程中负荷变动少，容易控制，缺点是动作区域狭窄。

(3) 球坐标型和圆柱坐标结构相比，这种结构更为灵活，占地面积小，

工作空间大，但是移动关节不易防护。

(4) 关节型这类机器人由两个肩关节和一个肘关节进行定位，由两个或三个腕关节定向。其中一个肩关节绕铅直轴旋转，另一个肩关节实现俯仰，这两个肩关节轴线正交。肘关节平行于第二个肩关节轴线。这种结构动作灵活，工作空间大，在作业空间内手臂的干涉最小，结构紧凑，占地面积小，手爪可获得较高的线速度。目前中小型机器人多采用这种结构。

(5) SCARA 型这种机器人有三个旋转关节，其轴线相互平行，在平面内进行定位和定向。另一个关节是移动关节，完成垂直平面的移动。它特别适用于把针脚等零件插入孔穴的场合，在完成精密装配操作任务中能充分发挥其效益^[11, 12]。

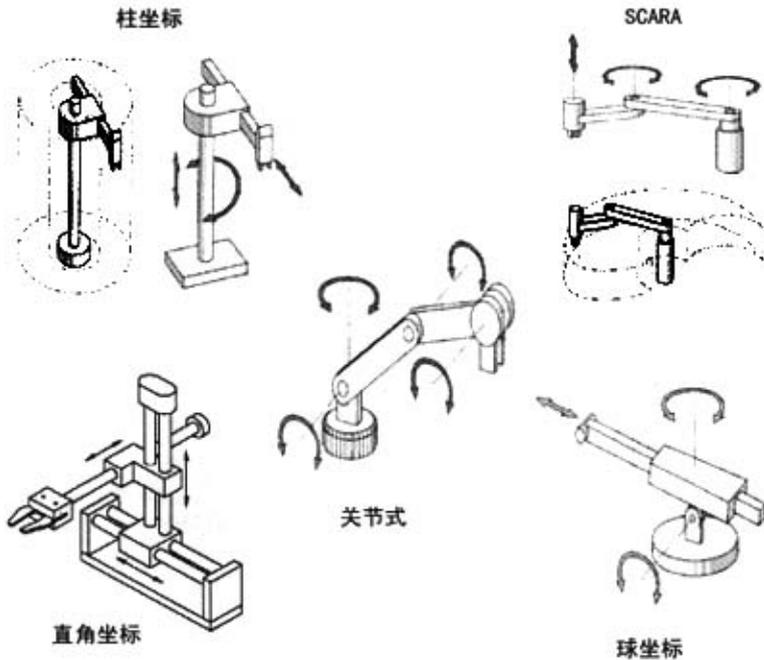


图 2.1 机器人的坐标形式

机器人的动态特性通常用质量、惯性矩、刚度、阻尼系数、固有频率和振动模态来描述。为了提高系统的动态特性，在机械结构设计时应该尽量减少质量和惯量，提高结构刚度和系统的固有频率，增加阻尼。本系统采用关节式五自由度的运动形式。关节型结构形式的机器人动作灵活、工作空间大、

在作业空间内手臂的干涉最小、结构紧凑、占地面积小、关节相对部位密封防尘。但关节型机器人的运动学原理较复杂，逆解困难，确定末端件的位置不直观，进行控制时计算量比较大。笔者设计的五自由度教学机器人的机械结构图如图 2.2 所示。该五自由度教学机器人包括：底部的腰回转、大臂的俯仰、小臂的俯仰三个位置自由度和小臂的回转、手腕俯仰两个姿态自由度和手爪抓取的动作，每个动作都用一个直流伺服电机加以驱动，因此这个机器人驱动系统需要六个直流伺服电机来实现其运动控制。

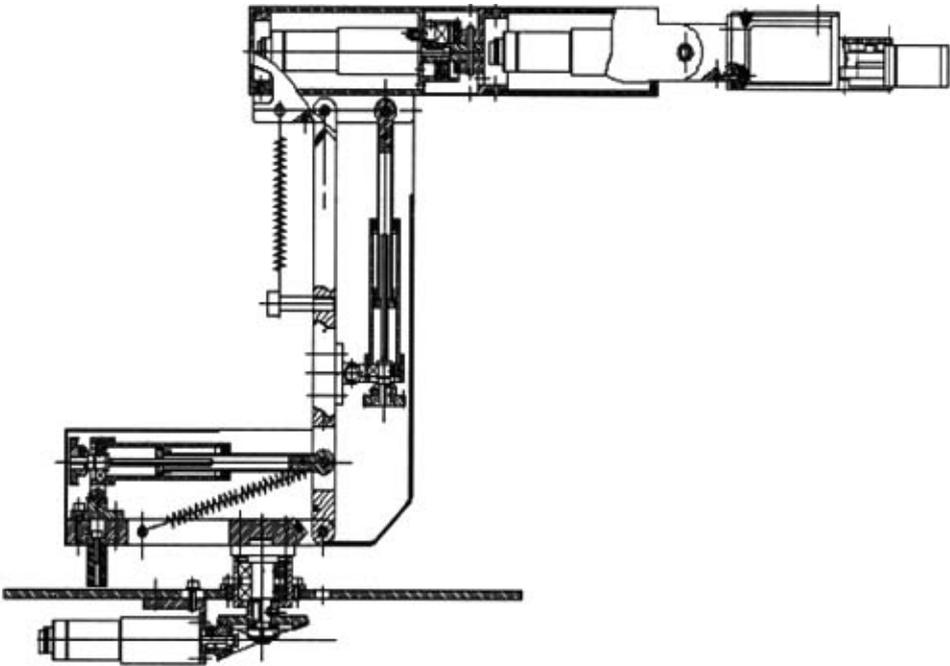


图 2.2 五自由度教学机器人结构图

五自由度教学机器人的各个关节机械结构的基本参数，材料选用硬铝合金，如表 2.1 所示。

表 2.1 机器人系统的基本参数

末端负载质量	1kg	
杆长	高度	300mm
	大臂长	260mm
	小臂长	260mm
关节转角范围	腰回转	$\pm 90^\circ$
	大臂俯仰	$\pm 45^\circ$
	小臂俯仰	$\pm 45^\circ$
	小臂回转	$\pm 90^\circ$
	手爪俯仰	$\pm 90^\circ$
手爪开合宽度	1mm~5mm	
定位精度	末端重复定位精度 ± 5 mm	

2.1.2 驱动方案

驱动方案包括确定机器人的驱动方式和驱动元件。机器人驱动方式有电机、液压和气动三种。电动系统适合于中等负载、动作复杂、运动轨迹严格的场合。关节驱动方式有直接驱动和间接驱动两种。关节直接驱动机器人(DDR)将电机直接做在关节上。目前大部分机器人的关节是间接驱动,即电机通过减速器装载以关节轴相连。

在选择机器人的制造材料时要综合考虑材料的强度、刚度、比重、弹性、抗振性、可加工性、外观及价格等。常用的材料有碳素结构钢、合金结构钢、铝、铝合金等等。本系统采用铝合金材料。

在机器人的设计中,可选择通用的标准零部件,如伺服电机、减速器、传感器和手爪等。对一些专用的部件则需自行设计,如底座、手臂和交叉滚子轴承^[13]。

本体采用直流伺服电机驱动,位置、速度检测系统由电机的自带的相对式正交编码器和非接触式霍尔行程开关构成。自研的电机驱动器上带有各种安全保护与报警接口,可用于保护电机和减速器。各关节上采用的电机及驱

动型号如表 2.2 所示。

如图 2.2 所示。各个关节的采用的驱动方式分别为：

- (1) 底部回转的关节，通过一对锥齿轮进行直角传动，传动比为 3.4；
- (2) 大臂俯仰关节，通过一对传动比为 1 的圆柱直齿轮传动，然后驱动丝杠螺母进行直线传动，推动大臂，为减少负载的影响，保证运动的平稳性，加装平衡弹簧；
- (3) 小臂俯仰关节，通过一对传动比为 1 的圆柱直齿轮传动，然后驱动丝杠螺母进行直线传动，推动小臂，为减少负载的影响，保证运动的平稳性，加装平衡弹簧；
- (4) 小臂回转关节，由电机加减速器直接输出驱动；
- (5) 腕部回转关节，由一对锥齿轮直角传动实现；
- (6) 手爪的开合，由丝杠螺母副和一连杆机构驱动。

表 2.2 机器人系统使用部件

关节部位	电机	减速器	编码器
腰回转	2342S024CR	246:1	IE2-512
大臂俯仰	2342S024CR	3.71:1	IE2-512
小臂俯仰	2342S024CR	3.71:1	IE2-512
小臂回转	2342S024CR	246:1	IE2-512
手爪俯仰	2342S024CR	246:1	IE2-512
手爪开合	24V DC Motor	—	—

2.2 控制系统方案

2.2.1 控制方式

五自由度教学机器人包括：底部回转、大臂俯仰、小臂俯仰、小臂回转、手腕俯仰五个自由度和手爪抓取的动作，每个动作都用一个直流伺服电机加以驱动，机器人共需要六个电机来实现其运动控制。就每个电机的位置伺服控制而言，是一个典型的位置伺服系统，因此采用典型的 PI 速度内环加 PID

位置外环的控制策略进行控制。在实际系统中,采用单片机数字控制的方法,PWM 信号驱动 H 桥的方式作为控制器的输出驱动电动机工作,以电机尾部的相对式编码器信号的作为反馈回路。整个系统需要 6 路独立的 PWM 输出接口,驱动五路伺服电机以及控制手爪开合电机,5 路正交编码器输入接口。

机器人的控制系统按其控制方式主要有分散控制和集中控制两种。各种控制形式的优缺点:

1. 分散控制

分散控制也称分级控制。分级控制的特点是:功能分散、任务分散、维护容易,适用于回路较多的控制系统。

上位机负责发送控制命令、数据输入、状态显示,接收下位机反馈回来的信息,经过轨迹规划、控制算法,向下位机发送控制命令。应满足:

(1) 上位机应该具有比较强的运算能力和数据处理能力、具有良好的人机接口和操作界面;具有一定数量的输出接口、输入接口,以及通信接口。上位机可以采用 ARM, DSP 等微处理器,复杂系统可以采用工控 PC, PC104 等系统机。

(2) 下位机具有良好的实时性。快的运算速度,在一个采样周期(毫秒量级)内要完成单个伺服回路的信号采样、滤波算法和控制算法,快速实时通信能力、丰富的输出接口(PWM 或者 DA)、输入接口(快速 A/D、编码器接口、数字 I/O)。下位机可以采用单片机、DSP 或者专用控制器。

2. 集中控制

集中控制与分级控制正好相反,由一台计算机完成整个系统的控制功能。轨迹规划、系统管理、伺服控制都有一台计算机完成。集中控制一般采用工业 PC 机, PC104, ARM, DSP 等数据处理能力强的计算机或者微处理器。控制计算机应满足一下特点:

(1) 满足比较强的运算能力和数据处理能力、具有良好的实时性、快的运算速度。在一个采样周期(毫秒量级)内完成系统的各个伺服回路的控制算法、信号采样、滤波算法、轨迹规划等;

(2) 具有良好的人机接口和操作界面;

(3) 具有一定数量的输入/输出接口,以及通信界面。

在五自由度教学机器人这个实例上，采用分级控制的方案，由 PC 机构成上位机，完成轨迹规划、系统管理的算法；下位机由一个或者几个单片机完成对六个关节的伺服控制^[14, 15]。

方案一：使用 6 个微处理器对 6 个电机进行分别控制，六个微处理器之间分别与上位机进行通信。这样每个微处理器上实现一路电机的闭环控制。优点是：在软件实现方面简单，成本比较低。缺点：系统比较庞大，耦合严重，造成与主机通讯协议复杂，可靠性比较低。

方案二：将 6 个电机进行统一控制，将 6 路的闭环控制算法在一个微处理器上实现。对这个微处理器的要求比较高，并且要求接口丰富，特别是 PWM 输出和 QEP 正交编码器接口。优点是：规模小，集成度高，可靠性高，并且和上位机单独通信，协议简单，可靠性高。缺点：成本较高。

经过比较两个方案的优劣，笔者采用方案二的控制方案。因此整个系统需要一个具有计算速度快、数据存储容量大、6 路独立 PWM 输出、6 路正交编码器输入、一定数量 I/O 的处理器。通过市场调研，发现没有带 6 路电机正交编码器输入接口的处理器。

经过一番实践，笔者摸索出一种全新的多自由度的电机伺服方案，选用 ATmega128+FPGA(现场可编程门阵列)的硬件构架来实现对整个机器人的六个关节的伺服控制，其总体的系统结构如图 2.3 所示。

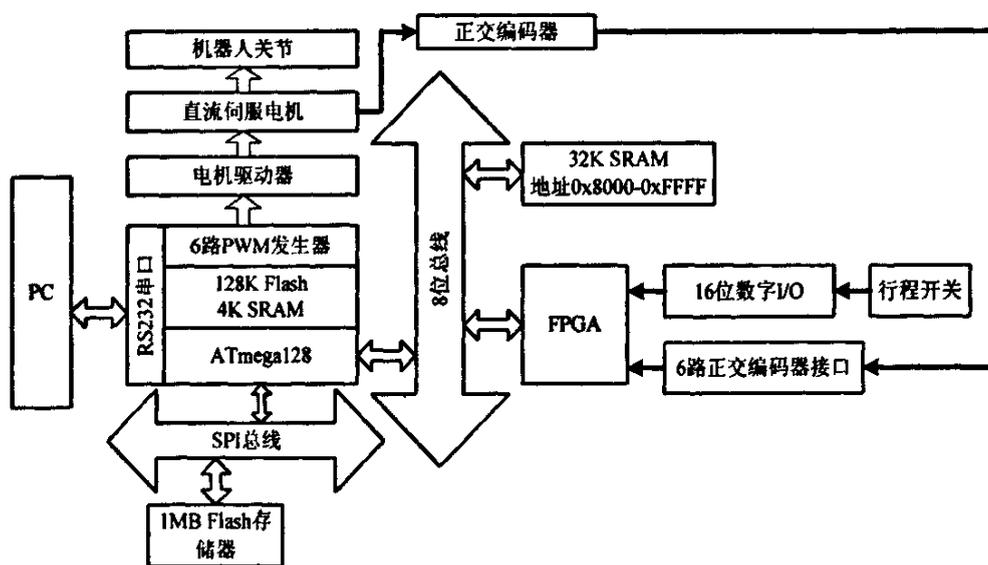


图 2.3 硬件系统结构图

2.2.2 工作原理

整个系统是由 ATmega128 高性能单片机、功率放大模块、存储模块、通信模块、FPGA 模块等组成。其中单片机主要负责控制算法的计算，功率放大模块实现 PWM(脉宽调制)信号的放大及电机的过流保护，存储模块实现示教再现数据的实时记录，通信模块负责与上位机进行通讯，FPGA 模块实现六路正交编码器接口、I/O 扩展、单片机中断扩展及单片机外部地址总线的地址译码。其系统的特点有：

- (1) ATmega128 为 CPU，当外部时钟为 14.7456MHz 时，14.7456MIPS 的运算能力^[16]；
- (2) 32K 的外部 SRAM，提供程序的运行数据空间；
- (3) 1MB 的串行 Flash 存储器，存放海量数据；
- (4) FPGA 实现 6 路正交编码器接口、16 位的数字 I/O 等数字电路；
- (5) 6 路独立的 PWM 输出隔离，通过 H 桥功率放大，以驱动电机；
- (6) 通过 RS232 通讯接口与上位机通讯，简单，实用。

在控制系统运行的过程中，用单片机内部的定时器产生周期为 5ms 的采样时间中断，以使采样时间任务由挂起状态进入运行状态。

在采样时间任务里，通过数据/地址总线读取 FPGA 内部的各个关节对应的正交编码器计数器寄存器，经过辨向，得到各个关节对应的电机当前位置，然后由单片机进行位置及速度闭环伺服控制算法得到输出的控制量，即 PWM 的占空比，经过功率放大，以驱动电机，在这过程中驱动模块的电流采样环节对电机的电枢电流进行实时的检测，当电流超过规定的阈值时，关闭驱动模块以保护电机。

单片机通过串行接收中断对上位机的命令进行检测，以实时的响应来自上位机的各种命令，并做出应答。同时当六个电机的某一个发生过流现象时、或当机器人关节超过极限行程时等错误状态时，通过发送错误代码向上位机进行报警。

2.3 本章小结

本章主要介绍了机器人机械系统的总体结构以及各个关节的驱动方式，对几种方案作了分析对比选择了 ATmega128+FPGA 的控制系统硬件构架，同时对课题所研究的五自由度教学机器人的控制系统组成及其工作原理作了总体的介绍。

第3章 控制系统的硬件设计

控制系统的硬件由主控单片机 ATmega128、功率放大模块、FPGA 外设扩展模块、通讯模块和存储器扩展组成。本章将详细的介绍每一个模块的组成及功能。

3.1 功率放大模块的设计

直流电动机具有良好的起、制动性能，宜于在大范围内平滑的调速，在许多需要调速或快速正反的电力拖动领域中取得了广泛的应用。直流电动机转速和其他参量之间的稳态关系可以表示为：

$$n = \frac{U - IR}{K_e \Phi} \quad (3-1)$$

式中 n —转速(r/min);

U —电枢电压(V);

I —电枢电流(A);

R —电枢回路总电阻(Ω);

Φ —励磁磁通(Wb);

K_e —由电机结构决定的电动势常数。

在上式中， K_e 是常数，电流 I 是由负载决定的，因此调节电动机的转速可以有三种方法：

- (1) 改变电枢供电电压 U ；
- (2) 减弱励磁磁通 Φ ；
- (3) 改变电枢回路的电阻 R 。

对于要求在一定范围内无级平滑调速的系统来说，以调节电枢供电电压的方式为最好。改变电阻只能有级调速；减弱磁通虽然能够平滑调速，但调速的范围不大，往往只能调压的方案。因此，自动控制的直流调速系统往往以变压调速为主^[7]。

调节电枢电压需要专门的可控直流电源，生产中常用的可控的直流电源有以下三种：

(1) 旋转变流机组，用交流电动机和直流电机组成机组，可获得可调的直流电压。

(2) 静止式可控整流，用静止式的可控整流器得到可调的直流电压。

(3) 直流斩波或脉宽调制，产生可变的平均电压。

由于前两种适用于大功率的电机控制，并且设备庞大，成本昂贵，而教学机器人属于小功率驱动，因此本系统的驱动方案选用 PWM(脉宽调制)的方式作为功率驱动的方式。用功率放大模块将来自单片机的 PWM 信号，进行放大，以用来驱动直流伺服电机。

3.1.1 PWM 电机驱动电路设计

驱动系统采用 PWM（脉宽调制）的方式驱动电机，把恒定的直流电源调制成频率一定、宽度可变的脉冲电压序列，从而可以改变平均输出电压的大小，以调节电机的转速。在比较几种 PWM 驱动的方式后，本文采用了 H 桥结构的双极性可逆的驱动电路。该电路有以下的优点：

(1) 电流一定连续；

(2) 可使电动机在四象限运行；

(3) 电动机停止时有微振电流，能消除静摩擦死区；

(4) 低速平稳性好，系统的调速范围宽；

(5) 低速时，每个开关器件的驱动脉冲仍较宽，有利于保证器件的可靠导通。

双极性的缺点是：在工作过程中，4 个开关器件可能都处于开关状态，开关损耗较大，而且在切换时可能发生上、下桥臂直通事故。

单个电机的 PWM 功率驱动电路原理图如图 3.1 所示。在功率驱动系统中，从单片机输出的 PWM 信号，经过一个 74F04 反向器变成两路相位相反的 PWM 信号。由于功率端使用 PWM 驱动电机，电流变化较大，如果与数字端共地，将严重干扰数字控制系统。因此用高速光隔离数字控制端和功率控制端(由于 PWM 的信号频率大约在 10kHz~20kHz 之间，一般的光隔

的开关的速度已不能满足要求), 提高系统的抗干扰性。PWM 信号经过光隔后, 驱动 IR2112 半桥驱动芯片。IR2112 内部内建了延时电路, 以保证上下桥臂不能同时导通。IR2112 驱动 4 个 IRF540 构成的 H 桥, 以驱动电机。IRF540 是一种高速开关的 CMOS 管, 持续工作电流 23A, 导通电阻仅 44 毫欧姆, 管压降很小, 并内自带续流二极管^[10]。

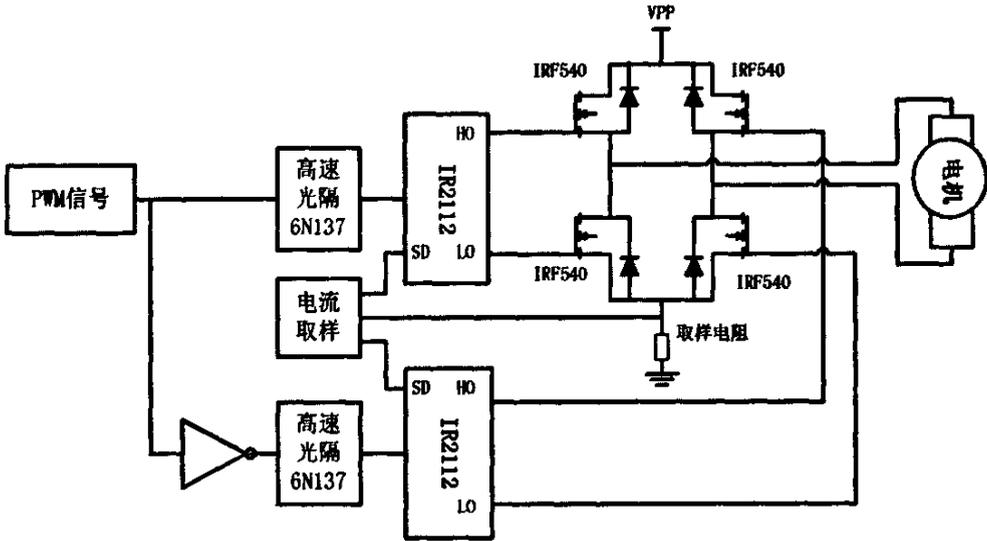


图 3.1 功率放大模块结构图

3.1.2 电流保护电路设计

由于整个系统所选用的电机属于小惯量高速电机, 过流能力差, 当电机发生堵转时, 电机的反电动势等于零, 造成电枢回路的电流急剧上升, 如果不将制止, 将发生电机烧毁的事件。

因此为了保护电机和伺服驱动器, 驱动系统增加了电机电枢回路电流保护电路, 如图 3.1 所示, 在 H 桥底部接 0.5 欧姆的电流取样电阻, 以采样 H 桥的电流。但是, 由于选用 PWM 的形式驱动电机, 因此采样电阻得到的电压是频率为 PWM 频率的脉动的电压信号, 变化较大, 不适合做使能控制信号。须对此电压经过滤波、放大、比较才能控制 IR2112 的使能端, 以起到过流保护的作用, 具体电路如图 3.2 所示。

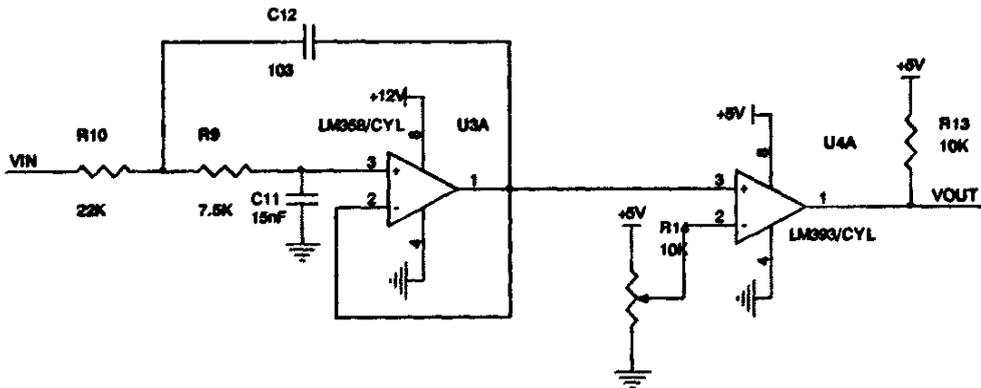


图 3.2 电流保护电路

其中 LM358 是单电源宽电压供电的 8 脚的双运算放大器，LM393 是集电极开路的模拟比较器。

电路原理：LM358 中的一个运算放大器配合外围的电阻、电容元件构成一个二阶的同相输入的巴斯特沃低通滤波器，增益为 1，带宽为 1kHz，其主要作用有：

- (1) 提高采样的输入阻抗。
- (2) 滤去 PWM 的高频干扰，得到一个相对平稳的电压值。

LM393 将运放得到的电压与设定的电压进行比较，若超过设定的电压，比较器输出高，使 IR2112 使能端失效，从而关断 H 桥的四个 CMOS 管，从而停止电机，达到限流，保护电机的目的。实践中由于功率的电源和地的干扰比较大，对运放和比较器干扰较大，因此设定的电压比理论计算值大，并在阈值电压处，存在斩波现象，属于正常情况。在理论上滤波器采样的电压有一定的滞后，导致对电流的滞后(毫秒量级)，但直流电机具有一定的过流能力，而且机械系统的响应频率较低(在 10Hz~100Hz 之间)，在实际中不影响使用，并取得了良好的电流保护功能^[19]。

3.2 基于 FPGA 的电机伺服模块的设计

伺服运动控制器的单片机除了需要实时接受并解释上位机的控制指令，计算并输出关节电机的控制信号外，还得实时监控机器人系统的状态转换，

为了不加重 CPU 的工作负荷(以往通过单片机中断软件计数的方法实现编码器计数,精度低,而且在编码器频率高时,严重干扰程序的正常执行),同时又保证整个伺服运动控制器的实时高效性能,决定将关节电机的光电位置编码器的反馈脉冲处理和计数电路、开关量检测电路连同外部设备地址译码电路,以及系统的复位信号综合电路等模块在高速的可编程逻辑器件中通过软件编程的方式加以实现。

FPGA 是现场可编程门阵列(Field Programable Gate Array)的简称,它基于查找表(Look Up Table)技术以及 SRAM 工艺,需要外挂配置用的 EPROM。FPGA 与 CPLD(复杂可编程逻辑器件)一样都是可编程逻辑器件,都是在 PAL, GAL 等逻辑器件的基础之上发展起来的。同以往的 PAL, GAL 相比, FPGA/CPLD 都具有体系结构和逻辑单元灵活、集成度高以及适用范围宽等特点。这两种器件兼容了 PLD 和通用门阵列的优点,可实现较大规模的电路,编程也很灵活。与门阵列等其它 ASIC(Application Specific IC)相比,它们又具有设计开发周期短、设计制造成本低、开发工具先进、标准产品无需测试、质量稳定以及可实时在线检验等优点,因此被广泛应用于产品的原型设计和产品生产。

FPGA 中采用的查找表(Look Up Table)技术简称为 LUT,其本质上就是一个 RAM。目前 FPGA 中多使用 4 输入的 LUT,所以每一个 LUT 可以看成是一个有 4 位地址线的 16x1 的 RAM。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后,PLD/FPGA 开发软件会自动计算逻辑电路的所有可能的结果,并把结果事先写入 RAM,这样,每输入一个信号进行逻辑运算就等于输入一个地址进行查表,找出地址对应的内容,然后输出即可。

综合考虑到要求可编程逻辑器件实现的功能电路较多,需要具有大量的输入/输出引脚,以及便于以后的控制功能扩展,因此控制系统的 FPGA 决定采用 Altera 公司的 ACEX1K 系列的 FPGA 芯片 EP1K30TC144。该芯片具有很高的逻辑密度,拥有门 30000 个,拥有 1728 个可编程的基本逻辑单元(LES)以及 102 个用户可编程的通用 I/O 引脚;它的功耗低,其内核供电电压为 2.5V,通用 I/O 块的供电电压为 3.3V,且能与 2.5V, 3.3V 和 5V 的外部设备信号兼容。

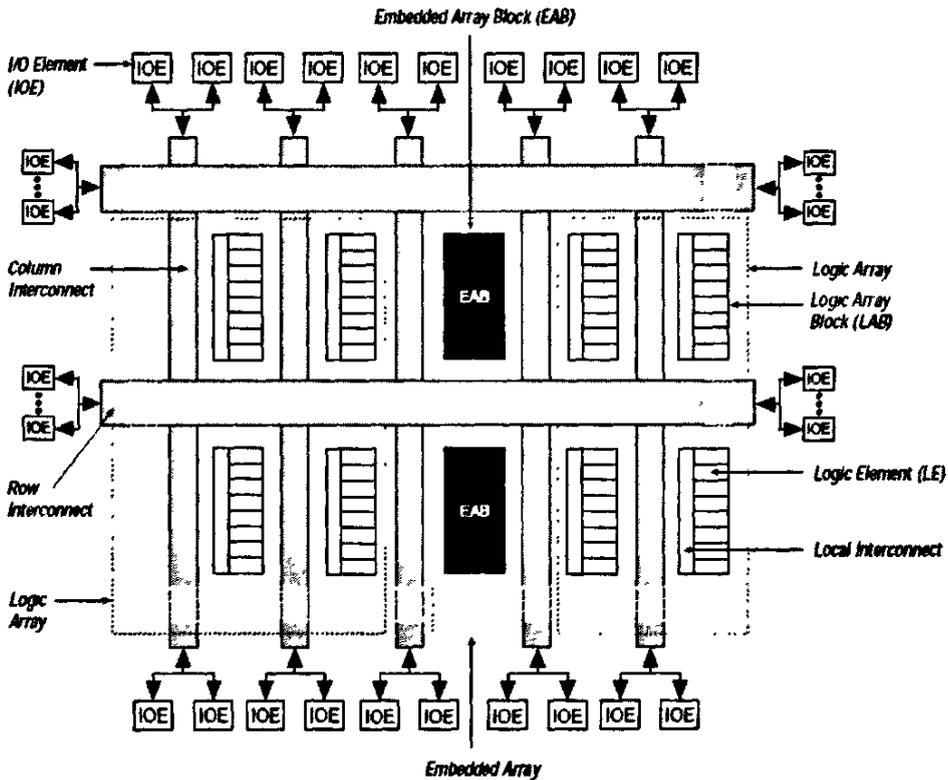


图 3.3 Altera ACEX1K 芯片内部结构

ACEX1K 系列 FPGA 的结构如图 3.3 所示。内部主要包括 LAB(逻辑阵列块)、I/O 块、RAM 块(未表示出)和可编程行/列连线等部分。在 ALEX 中，一个 LAB 包括 8 个逻辑单元(LE)。LE 是 ACEX 芯片实现逻辑的最基本结构，每个 LE 包括一个 LUT，一个触发器和相关的逻辑，片内的 RAM 即可用于实现双口 RAM，也可用于实现 FIFO，还可以当做 ROM 使用，具体情况如图 3.4 所示。

与该 FPGA 配合使用存放配置文件的外部存储器是 Altera 公司的 EEPROM 芯片 EPC2，系统上电后，将 PROM 中存储的程序加载到 FPGA 中，使 FPGA 按照设定的程序工作。

本系统的 FPGA 设计采用 QuartusII，其是 Altera 公司的新一代 FPGA 开发工具，支持 APEX 系列、Excalibur、Mercury、Stratix 以及 Cyclone 系列器件。为了缩短设计周期和降低设计复杂度，QuartusII 含有工作组计算、集成逻辑分析功能、EDA 工具集成、多过程支持、增强重编译和 IP 集成等特性^[20]。

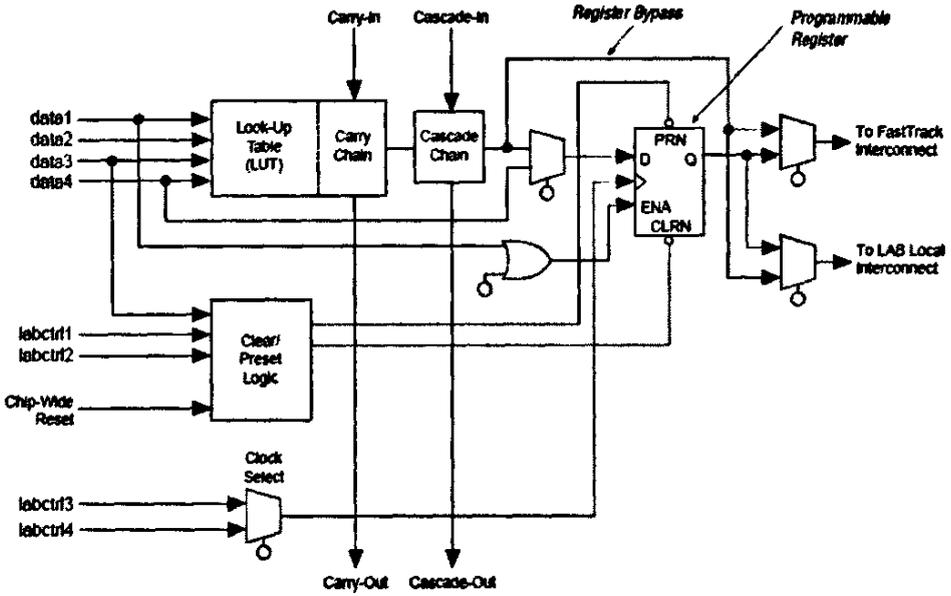


图 3.4 逻辑单元 LE 内部结构

在伺服控制器中，FPGA 主要由以下几个模块组成如图 3.5 所示。在机器人控制系统中 FPGA 主要是用来实现 6 路正交编码器接口和用于行程开关检测的 I/O 接口及单片机中断接口的扩展等^[21]。

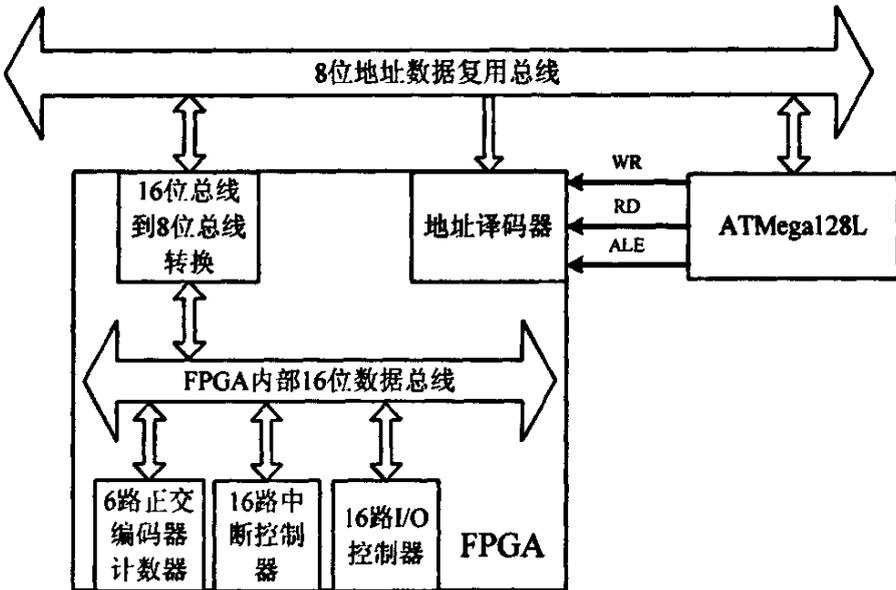


图 3.5 FPGA 内部结构

正交编码器模块，相对式编码器的接口，对编码器器信号做滤波、辨相及计数；

16 路 I/O 控制器，16 位并行 I/O 口扩展，完成多路 I/O 信号的同时读取，如绝对式编码器；

16 路中断控制器，扩展单片机的中断，用于 FPGA 内部或外部的中断申请；

总线变换模块，完成 FPGA 内部 16 位总线与单片机 8 位总线的变化；

地址译码模块，对单片机的地址进行译码，完成 FPGA 内部模块的选通控制。

3.2.1 正交编码器接口的设计

本系统采用的直流电机内部集成了增量式旋转编码器，输出为两个相位相差 90 度的方波信号，电机一转输出 512 个脉冲。传统的单片机处理编码器信号的方法是：通过中断计数，然后运用一定的算法进行辨向，也有通过逻辑电路进行辨向的。这种方法的缺点是：中断计数的使用使单片机疲于进入中断，退出中断，浪费了大量的宝贵的 CPU 时间，不能很好的发挥 CPU 的计算能力的优点，是对 CPU 资源的浪费。

在五自由度教学机器人电机伺服系统中采用 FPGA 代替单片机对电机编码器进行计数，单片机通过地址总线对 FPGA 内部寄存器进行访问。从而得到电机的速度及位置^[22-24]。

3.2.1.1 正交编码计数原理

电机编码器为 A、B 两相在相位上相差 90 度的脉冲信号。设当 A 超前 B 时，电机正转，编码器值增加，当 A 滞后 B 时，电机反转，编码器值减小。将编码器 A、B 相信号状态，记做状态 AB，当 A、B 相信号发生变化时，得出编码器四倍频时的状态转换图如图 3.6 所示。

当 AB 状态从：00→10→11→01→00 时，编码器计数值均是加一；
当 AB 状态从：00→01→11→10→00 时，编码器计数值均是减一。

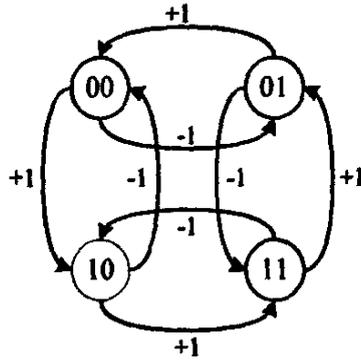


图 3.6 正交编码器计数原理图

3.2.1.2 正交编码计数器模块的设计及仿真

使用 Altera 公司提供的 EDA 工具 Quartus II 5.1, 使用 Verilog HDL 硬件描述语言对该状态机进行描述, 图 3.7 描述了正交编码计数模块的内部接口结构及接口情况。

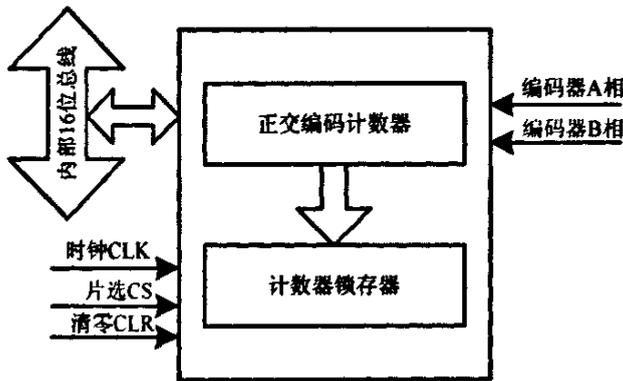


图 3.7 正交编码器计数模块结构及接口

ATmega128 通过地址总线访问此计数器。由于电机的最高转速为 8000r/min, 系统的采样时间为 5ms, 计算在采样时间内, 编码器四倍频后的最大输出值为 342 个, 因此采用 16 位计数器进行计数, 而 ATmega128 是 8 位总线, 因此在发生总线读写时, 须将 16 位的计数器值锁存。图 3.7 中 A、B 为电机的编码器信号, CLK 为时钟信号, CS 为片选信号, 高有效, CLR

为清零信号，高有效。六个计数器对应 ATmega128 的外部地址为 0x1100~0x110A。

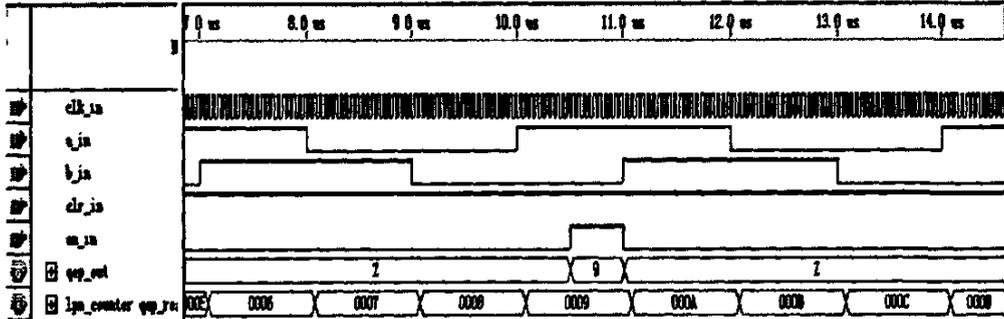


图 3.8 正交编码计数器仿真验证图

在 Quartus II 5.1 对正交编码器计数模块进行仿真分析，如图 3.8 所示。

- 图中 clk_in: 时钟信号;
- a_in,b_in: 编码器信号;
- clr_in: 编码器清零信号;
- en_in: 计数器选通信号;
- qep_out: 计数器选通时的输出;
- lpm_counter: 内部计数器的计数值。

3.2.2 多路中断控制器的设计

由于在 FPGA 内部集成了众多的模块，有的模块当检测到状态改变时，需要单片机进行处理，做出响应，例如：I/O 模块的某个 I/O 的电平发生变化、上升沿或下降沿时，需要 CPU 做出响应。

3.2.2.1 中断模块的设计

中断模块包括三个寄存器：中断屏蔽寄存器、中断控制寄存器 1、中断控制寄存器 2、中断挂起寄存器。它接收 16 个固定优先级的中断，每个中断可以设定为电平变化触发、上升沿和下降沿触发。中断屏蔽寄存器设定每个中断，中断挂起寄存器记录了系统没有响应的中断，中断模块通过中断屏蔽

寄存器的值，发出电平变化的中断请求，其结构如图 3.9 所示。图中 INT0—INT15 为外部中断输入，A0, A1 为访问寄存器地址，CS 为片选，INT 为中断请求输出（低电平申请中断），其余为读写控制信号。INTCR0 控制输入中断方式：电平或边沿。INTCR1 控制输入中断在电平方式时的高电平或低电平，在边沿方式时的上升沿或下降沿。当发生中断时，模块根据中断屏蔽寄存器的设置，置中断挂起寄存器。主机响应对应中断后，向对应为写 ‘1’，即可清中断挂起标志。对应 ATmega128 片外地址为：INTCR0 (0x1130)，INTCR1 (0x1132)，INTMASK (0x1134)，PEND (0x1136)。

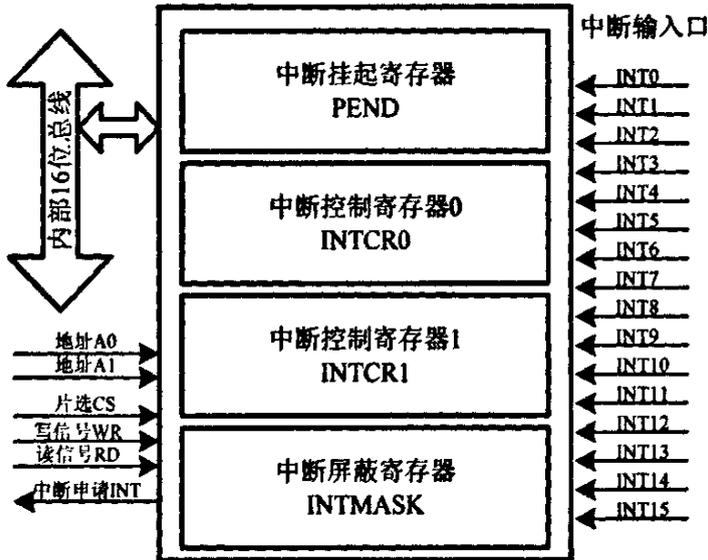


图 3.9 中断模块结构

3.2.3 多路 I/O 控制器的设计

由于五自由度教学机器人是一个复杂的系统，整个系统需要大量的开关量的检测，而 ATmega128 单片机的 I/O 口的功能强大，往往复用了大量其他的功能，而且数量比较少。FPGA 有大量的 I/O 口剩余，虽然对 I/O 的扩展有多的方法，但从降低系统复杂度、提高可靠性以及经济性的方面考虑，笔者利用 FPGA 剩余的资源，构造了一个 16 位的 I/O 口模块，用于对开关量的检测，其使用方法和 ATmega128 内部的 I/O 口模块的使用一样^[24]。

其结构如图 3.10 所示。由三个寄存器组成：PORT (I/O 输出寄存器)，DDR (I/O 方向寄存器)，PIN (I/O 引脚电平寄存器)。DDR 的位控制对应 I/O 引脚的输入输出，当 DDR 为高时，I/O 口设置为输出，这时 PORT 口的对应位控制 I/O 输出的高低电平，‘1’为高，‘0’为低。当 DDR 为低时，I/O 口设置为输入，这时对应 PIN 的值即引脚的电平值。其中 PIN 寄存器输出带锁存器，以保证在读时值不发生变化。其读写与上述模块相同。对应 ATmega128 片外地址为：PORT (0x1120)，DDR (0x1122)，PIN (0x1124)，其中 PORT 和 DDR 可读可写，PIN 为只读。

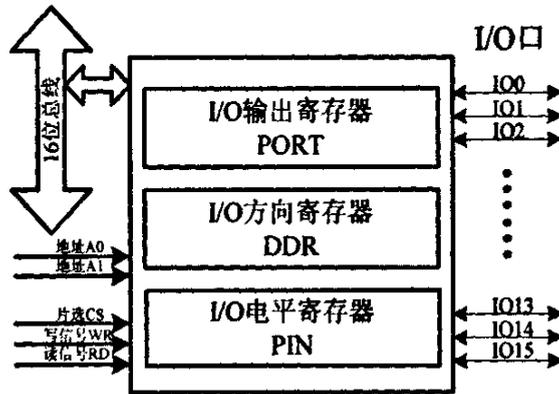


图 3.10 16 位数字 I/O 口结构

3.2.4 FPGA 与单片机之间的接口设计

以上的几个模块均是连接在 FPGA 内部的 16 位数据总线上，而控制系统选用的单片机 ATmega128 是 8 位地址/数据复用总线，就是说，单片机对 FPGA 内各个的寄存器的访问必须严格按照 ATmega128 访问外部数据空间的时序进行数据访问。当 ATmega128 发生一次外部数据访问时，首先在 ALE 上出现高电平，半个 CPU 时钟周期后在 A[15:8]和 D[7:0]上出现地址，并在 ALE 下降沿稳定。如果写操作，在一个 CPU 时钟周期后出现 \overline{WR} 的写信号，并在 \overline{WR} 出现下降沿前，数据已经稳定地建立在 D[7:0]数据总线上，并且在 \overline{WR} 信号变高后再持续有效一段时间。如果是读操作，同样在一个 CPU 时钟周期后出现 \overline{RD} 的读信号，并要求外设在一定时间在数据总线后建立，然后读取数据，并且要求数据保持到读信号 \overline{RD} 变高以后。如果不能在规定的时间内

建立数据，意味着单片机读到不稳定数据，这种错误是十分严重。时序如图 3.11 所示，时间要求如表 3.1 所示。

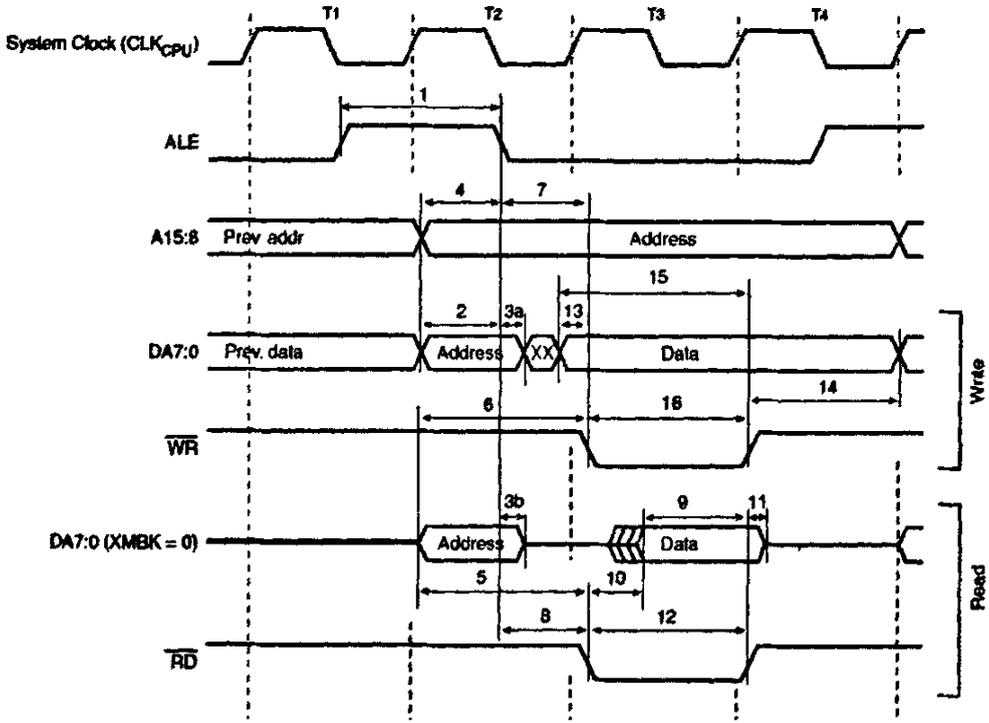


图 3.11 ATmega128 访问外部数据的时序图

表 3.1 外部数据存储特性参数表

符号	参数	最小值	最大值	单位
$1/t_{CLCL}$	振荡器频率	0	16	MHz
t_{LHLL}	ALE 脉冲宽度	$1.0t_{CLCL} - 10$		ns
t_{RLDV}	地址有效到 ALE 低	5		ns
t_{AVRL}	地址有效到 \overline{RD} 低	$1.0t_{CLCL} - 10$		ns
t_{AVWL}	地址有效到 \overline{WR} 低	$1.0t_{CLCL} - 10$		ns
t_{DVRH}	数据建立到 \overline{RD} 高	40		ns
t_{RLDV}	读信号低到数据有效		$1.0t_{CLCL} - 5$	ns
t_{RLRH}	读脉冲宽度	$1.0t_{CLCL} - 10$		ns
t_{WLWH}	写脉冲宽度	$1.0t_{CLCL} - 10$		ns
t_{DVWL}	数据有效到 \overline{WR} 低	$0.5t_{CLCL} - 20$		ns
t_{DVWH}	数据有效到 \overline{WR} 高	$1.0t_{CLCL}$		ns

本伺服控制系统主单片机选用的晶振为 14.7465MHz，得到总线操作时，各个主要时序参数，如下：

- (1) 系统时钟 14.7465MHz，即 68ns；
- (2) 地址锁存时间 58ns；
- (3) 读操作时，数据有效时间 40ns；
- (4) 数据建立到写有效时间 14ns；
- (5) 读信号宽度 58ns；
- (6) 写信号宽度 58ns；
- (7) 地址锁存有效到读/写信号有效 58ns。

只有满足上述的时间要求时，才能保证数据读取的正确性。

由于 FPGA 内是 16 位数据总线，对于 ATmega128 的 8 位总线的读写，设计了一个专门接口进行转换，并且为了保证读数据的正确性，在读操作中加入了数据锁存器，在进行第一次 8 位的读操作时，将 16 位数据锁存。

3.2.4.1 16 位总线与 8 位总线的变换模块

设计中将两次 8 位总线的操作合并一次 16 位总线操作。对于地址译码，读取数据采取偶地址对齐的方式，低字节在前，高字节在后，结构如图 3.12。

对于两次的 8 位总线的读操作，FPGA 内部的 16 位总线与 8 位总线变换模块进行如下操作：

在第一次读操作开始时，通过总线变换模块发出读信号，并将数据锁存在临时的 16 位寄存器中，并将低 8 位送到 8 位的数据总线；第二次读操作直接从 16 位的寄存器中取出高 8 位，送到 8 位的数据总线上。

对于两个 8 位总线的写操作，FPGA 内部的 16 位总线与 8 位总线变换模块进行如下操作：

在第一次写操作完成后，先将数据写入 16 位的临时写寄存器的低 8 位上；在第二次写操作后，先与原来的 8 位数据组成一个完整的 16 位数据，然后一次性将 16 位临时写寄存器的值通过内部的 16 位数据总线写入目标寄存器。

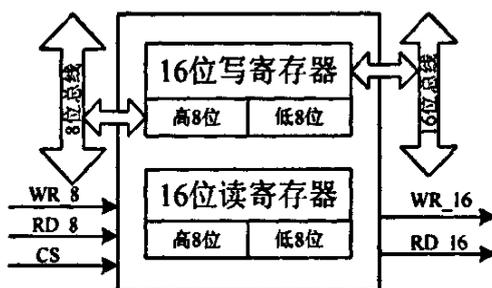


图 3.12 8 位总线与 16 位总线变换模块结构

3.3 主单片机控制系统的设计

五自由度教学机器人的主控单片机基于 RISC 高性能 ATmega128 单片机。ATmega128 单片机是高性能、低功耗的 AVR 8 位微处理器。AVR 内核具有丰富的指令集和 32 个通用工作寄存器。所有的寄存器都直接与算术逻辑单元 (ALU) 相连接，使得一条指令可以在一个时钟周期内同时访问两个独立的寄存器。这种结构大大提高了代码效率，并且具有比普通的复杂指令集微处理器高 10 倍的数据吞吐率。

ATmega128 具有如下特点: 128K 字节的系统内可编程 Flash(具有在写的过程中还可以读的能力, 即 RWW)、4K 字节的 EEPROM、4K 字节的 SRAM、53 个通用 I/O 口线、32 个通用工作寄存器、实时时钟 RTC、4 个灵活的具有比较模式和 PWM 功能的定时器/计数器(T/C)、两个 USART、面向字节的两线接口 TWI、8 通道 10 位 ADC(具有可选的可编程增益)、具有片内振荡器的可编程看门狗定时器、SPI 串行端口、与 IEEE1149.1 规范兼容的 JTAG 测试接口(此接口同时还可以用于片上调试), 以及六种可以通过软件选择的省电模式。空闲模式时 CPU 停止工作, 而 SRAM、T/C、SPI 端口以及中断系统继续工作; 掉电模式时晶体振荡器停止振荡, 所有功能除了中断和硬件复位之外都停止工作, 寄存器的内容则一直保持; 省电模式时异步定时器继续运行, 以允许用户维持时间基准, 器件的其他部分则处于睡眠状态; ADC 噪声抑制模式时 CPU 和所有的 I/O 模块停止运行, 而异步定时器和 ADC 继续工作, 以减少 ADC 转换时的开关噪声; Standby 模式时振荡器工作而其他部分睡眠, 使得器件只消耗极少的电流, 同时具有快速启动能力; 扩展 Standby 模式则允许振荡器和异步定时器继续工作。器件是以 Atmel 的高密度非易失性内存技术生产的。片内 Flash 可以通过 SPI 接口、通用编程器, 或引导程序多次编程。引导程序可以使用任何接口来下载应用程序到应用 Flash 存储器。在更新应用 Flash 存储器时引导 Flash 区的程序继续运行, 实现 RWW 操作。通过将 8 位 RISC CPU 与系统内可编程的 Flash 集成在一个芯片内, ATmega128 为许多嵌入式控制应用提供了灵活而低成本方案^[25-27]。

3.3.1 电源及复位电路的设计

这个系统由功率部分和数字控制部分组成, 由于功率部分属于强电驱动, 如果和数字部分统一供电, 会严重干扰数字控制部分, 引起单片机控制系统的复位。因此, 供电方式采用隔离供电的方案: 采用单电源+24V 供电, 通过 DC/DC DLM05-24S05 隔离供电给数字控制部分, 这样由功率端带来的干扰隔离了。功率部分采用+24V、+12V、+5V 供电, 数字控制部分采用+5V、+3.3V、+2.5V 供电。

复位电路采用 MAX809T, +3.3V 复位芯片, 为 ATmega128 提供复位时

序逻辑，完成上电复位。而 FPGA 中各个模块的的复位由 ATmega128 的 I/O 口控制，由于 FPGA 采用的上电加载程序的方法加载程序，完成 FPGA 程序加载后，FPGA 会将 INIT_DONE 的电平拉高，以通知单片机。因此，FPGA 内部寄存器的复位，只能在 FPGA 程序加载完成后进行，而不能采用上电复位的方式^[28, 29]。

3.3.2 主单片机与 FPGA 的接口设计

FPGA 作为主单片机的一个外设，主单片机通过地址/数据总线访问 FPGA 的内部寄存器，实现数据交换。并且通过 FPGA 内部的地址译码模块实现对其他外部设备的地址译码，两者的接口关系结构如图 3.13 所示。

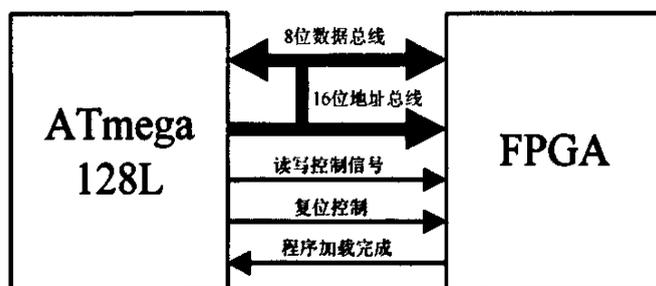


图 3.13 ATmega128 与 FPGA 的接口

在单片机程序设计部分，对 16 位寄存器的访问需要注意以下问题：

(1) 两次 8 位数据访问中，需要禁止中断，防止中断插入，破坏数据访问的时序；

(2) 访问顺序必须严格按照 FPGA 中定义的顺序，即先访问低地址的字节，然后访问高地址的字节。如果用 C 语言进行编程，需要对编译器产生的汇编代码进行校验，保证产生的代码符合顺序，必须偶地址对齐；

(3) 单片机上电后，必须等待 FPGA 程序加载完成，并在 FPGA 加载完成后对 FPGA 进行复位，使 FPGA 内部的寄存器有确定的值，I/O 口的引脚有确定的电平。

3.3.3 通讯电路设计

通讯电路采用标准的 RS232 通信，RS232 串行通信接口标准是指串行通信接口与外设的信号联机标准。通常有 3 种：RS-232C、RS-422 和 20mA 电流环。我们选择的 PC 机配置的是 RS-232C 标准接口。RS-232C 的定义包括电气特性、机械特性、功能特性等。若不使用硬件流量控制，则只需利用 3 根信号线，便可以做到全双工的传输作业^[30, 31]。

通信的波特率为 115200bit/s，1 位停止位，8 位数据，无奇偶校验。系统中用 MAX3232 将单片机的 TTL 电平转换成 RS232 的逻辑的电平与上位机进行通讯，硬件电路如图 3.14 所示。

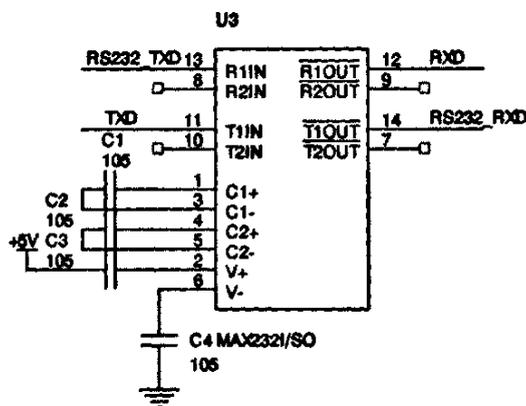


图 3.14 通讯电路图

3.3.4 存储器扩展

由于在本系统的软件编程采用 uC/OS II 实时操作系统作为软件框架，因此对数据存储器进行校验，因此，在系统原有的 4KB 的数据存储器的基础上，扩展了一片 IS61LV256，容量 32KB 的数据存储空间。在地址空间的分配上，从 0x1100~0x7fff 为 FPGA 片内寄存器和其他外设的地址空间，0x8000~0xffff 为 32KB 的 SRAM 的数据空间。

在机器人运动的过程中，需要对五个伺服关节的转角进行实时的记录。用 long 型的变量存储一个关节的转角，那五个关节的存储量是 20 个字节，

加上 2 个字节的索引。一共需要 22 个字节来描述一个关节的转角。若进行示教再现中，以 1s 记录一个示教点，若进行长时间的示教，所需的数据存储量巨大。因此在系统中扩展了一块 AT45DB081B，此芯片 1Mbit 的，基于 SPI 总线的串行 FlashROM，来完成各个角度数据的存储的缓冲。

3.3.5 AD 采样电路设计

本系统需要对模拟信号进行采集，利用 ATmega128 内置的模数转换器实现对模拟量的转换。ATmega128 的模数转换器的参数如下：

- (1) 10位精度，0.5LSB的非线性度， ± 2 LSB的绝对精度；
- (2) 13~260 μ s的转换时间，最高分辨率时采样率高达15kSPS；
- (3) 8路复用的单端输入通道，7路差分输入通道；
- (4) 2路可选增益为10x与200x的差分输入通道。

为了提高 AD 的输入阻抗，并且滤去一部分高频噪声，在 AD 输入的前级添加了正向端输入的增益为 1 的比例放大电路，并进行低通滤波，并用 5.1v 的稳压管保护的 AD 端口，电路如图 3.16 所示。

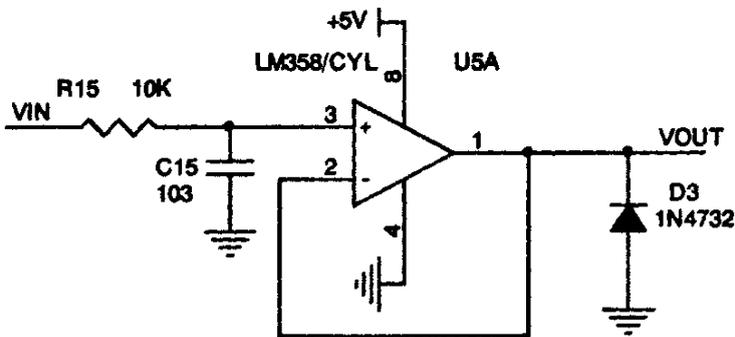


图 3.16 ADC 采样前级电路

3.4 本章小结

本章详细地介绍了五自由度教学机器人的控制系统的硬件构成。主要包括：功率放大模块、电机驱动电流保护的设计，FPGA 电机伺服模块及主控单片机的系统设计。详细地介绍 FPGA 内各个模块的内部实现的细节，以及各个模块之间进行通讯的方式以及 FPGA 和 ATmega128 之间的通讯。

第4章 控制系统的软计设计

4.1 数字PID控制

多年以来,在过程控制中,按偏差的比例(P)、积分(I)和微分(D)进行控制的PID控制器(亦称PID调节器)是应用最为广泛的一种自动控制器。它具有原理简单、易于实现、适用面广、控制参数相互独立、参数的选定比较简单等优点;而且在理论上可以证明,对于过程控制的典型对象“一阶滞后+纯滞后”与“二阶滞后+纯滞后”的控制对象,PID控制器是一种最优控制。PID调节规律是连续系统动态品质校正的一种有效方法,它的参数设定方式简便,结构改变灵活(如PI,PD,……)。长期以来广大科学技术人员及现场操作人员大量采用PID控制器,并积累了大量的经验^[22, 23]。

4.1.1 PID控制原理

连续线性系统的PID控制器为对误差的比例、积分和微分的控制,即:

$$u(t) = K_p[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt}] \quad (4-1)$$

或

$$u(s) = K_p \cdot e(s) + K_i \cdot \frac{e(s)}{s} + K_d \cdot s \cdot e(s) \quad (4-2)$$

式中: T_i 、 T_d 分别是积分常数和微分常数, K_p 、 K_i 、 K_d 分别比例系数、积分系数、微分系数。在计算机控制系统中用的数字控制系统。对(4-2)式进行离散化,令

$$\begin{cases} u(t) = u(kT) \\ e(t) = e(kT) \\ \frac{de(t)}{dt} = \frac{e(kT) - e(kT - T)}{T} \\ \int_0^t e(t) dt = T \cdot \sum_{j=0}^k e(jT) \end{cases} \quad (4-3)$$

式中 T 为采样时间，由(4-2)和(4-3)可得：

$$U(kT) = K_p \left\{ e(kT) + \frac{T}{T_i} \sum_{j=0}^k e(jT) + \frac{T_d}{T} [e(kT) - e(kT - T)] \right\} \quad (4-4)$$

式(4-4)为离散化后的位置式的 PID 算法^[34, 38]。在实际的系统中，需要对五个关节的电机进行位置伺服控制，其中保证在运动过程中平稳、具有一定的抵抗负载变化的能力，采用位置闭环做外环，速度闭环做内环的控制方案，结构如图 4.1 所示。

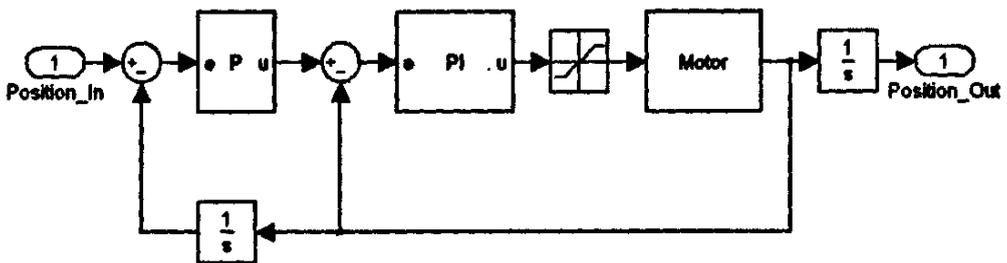


图 4.1 控制结构框图

图 4.1 中，位置闭环采用比例(P)控制，速度内环采用比例和积分(PI)控制，电机的反馈是电机的转速，经过一个积分环节后得到电机的位置。由于实际系统中伺服驱动器和电机的输出等物理量的输出存在物理最大值，因此在速度环的输出端，即控制量的输出端，增加了一个饱和环节，饱和的上下限位伺服驱动能够输出的最大电压，以保证控制的输出在物理允许的范围内^[37]。

4.1.2 PID 的程序流程

在程序中对每个关节的伺服电机的速度闭环和位置闭环建立对应的结构体对象，实现对机器人各个关节特征参数封装，这样对每个关节参数各自独立，互不干扰，并且提高了程序的可移植性、易读性和可靠性。

关节速度伺服闭环的 PID 结构体：

```
typedef struct
{
    INT16S  Qep_fb;           //编码器反馈值
    INT16S  PWM;             //输出的 PWM 值
    INT16S  ResetPWM;       //复位的 PWM 值
    INT16S  PWM_Up_Limit;   //PWM 上限
    INT16S  PWM_Down_Limit; //PWM 下限
    INT16S  Rst_Speed;      //复位是的速度
    INT16S  Speed_ref;      //速度给定
    INT32S  Pid_Output;     //PID 控制器输出
    INT16S  Pid_Output_Max; //PID 饱和上限
    INT16S  Pid_Output_Min; //PID 饱和下限
    INT16S  Pid_Ki;         //积分系数 Ki
    INT16S  Pid_Kd;         //微分系数 Kd
    INT16S  Pid_E2;        //偏差 E(k-2)
    INT16S  Pid_E1;        //偏差 E(k-1)
    INT16S  Pid_SumE;      //偏差的总和
    void    (*calc)();     //PID 控制算法
    void    (*init)();     //初始化函数
    void    (*updata)();   //PWM 输出
} SPEEDPID;
```

速度 PI 闭环程序流程如图 4.2 所示。

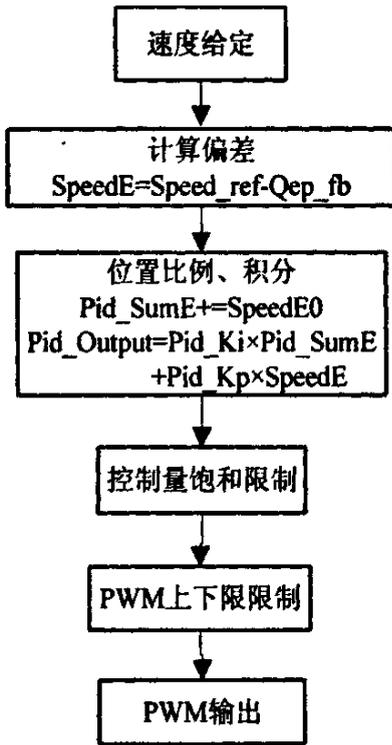


图 4.2 速度闭环流程

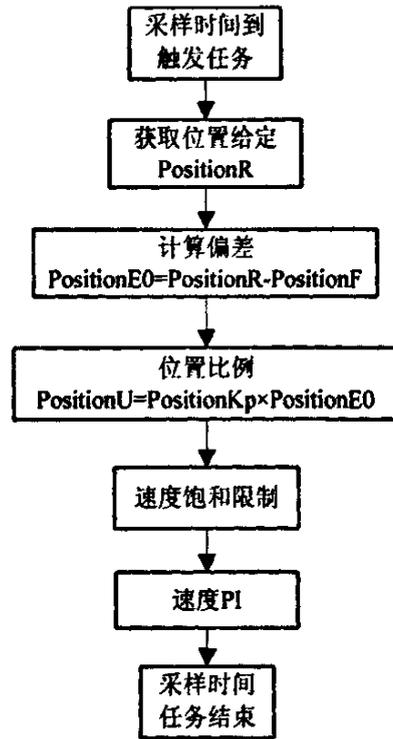


图 4.3 位置闭环流程

位置闭环的程序流程如图 4.3 所示。

关节位置伺服闭环的位置 PID 结构体：

typedef struct

```

{
    INT16S  TransGain;           //关节的传动比
    INT32S  PositionStart;      //关节的初始位置
    INT32S  PositionRightOver;  //关节的右极限位置
    INT32S  PositionLeftOver;   //关节的左极限位置
    SPEEDPID *Speed_reg;       //速度 PID 对象
    INT32S  PositionA;          //关节的当前的绝对位置
    INT32S  PositionR;          //电机的位置给定
    INT32S  PositionF;          //电机的位置反馈
    INT32S  PositionU;          //位置 PID 的输出
    INT16S  Speed;              //电机给定速度
}
    
```

```

INT32S PositionE1;          //偏差 e(k-1)
INT32S PositionE2;          //偏差 e(k-2)
INT32S PositionSumE;        //偏差的总和
INT16S PositionKp;          //比例系数 Kp
INT16S PositionKi;          //积分系数 Ki
INT16S PositionKd;          //微分系数 Kd
void (*calc);                //PID 控制算法
void (*update);              //控制器输出更新
}POSITIONPID;

```

由于 ATmega128 属于定点 CPU，因此在进行数值运算时，对原来的浮点的计算方法进行优化，在满足精度的前提下，最大可能的提高运算的速度。具体方法：用定点数的计算方法，代替浮点数的计算，例如：将除以 3，变成先乘以 341，然后在除以 1024 即右移 10 位，在实验中将两种方法进行对比，后者比前者快 100us，因为 AVR 为单指令周期的单片机，许多的指令为单周期执行，若晶振为 8MHz，那指令周期为 125ns，也就是说，后者比前者少了近 800 个指令周期。而在精度上，后者与实际的值只差 0.0003，完全能满足系统精度的要求。

4.2 uC/OS II 实时操作系统框架的构造

要把 uC/OS II 应用到实际的项目中来，首先是基于 uC/OS II 的诸多特点和优点，以及对 uC/OS II 内核深入理解基础上的。因此，下面将详细介绍实时操作系统内核^[30]。

与其他的操作系统不同，uC/OS II 其实只有一个内核，提供了任务管理、任务间的通信与同步、时间管理和内存管理等基本功能。

1. 任务管理

uC/OS-II 最多可以管理 64 个任务，这些任务通常都是一个无限循环的函数。在目前的版本中，作者保留了优先级为 0, 1, 2, 3, OS_LOWEST_PRIO_3, OS_LOWEST_PRIO_2, OS_LOWEST_PRIO_1, OS_LOWEST_PRIO 的任务，

所以用户可以拥有56个任务。uC/OS II提供了任务管理的各种函数，包括创建任务、删除任务、改变任务的优先级、挂起和恢复任务等。系统初始化时会自动产生两个任务：一个是空闲任务OSTaskIdle()，它的优先级最低为OS_LOWEST_PRIO，该任务只是不停地给一个32位的整型变量加1；另一个是统计任务OSTaskStat()，它的优先级为OS_LOWEST_PRIO-1，该任务每秒运行一次，负责计算当前CPU的利用率。

2. 任务间的通信与同步

对于一个多任务操作系统来说，任务间的通信与同步是必不可少的。uC/OS II提供了4种同步任务的方法，分别是信号量、邮箱、消息队列和事件。通过邮箱和消息队列还可进行任务间的通信。所有的同步对象都有相应的创建、等待、发送的函数。

3. 时间管理

uC/OS II要求用户提供一个称为时钟节拍的定时中断，以实现延时与超时控制等功能，该中断每秒发生10~100次。时钟节拍的频率是由用户控制的，任务申请延时或超过控制的计时基准就是该时钟节拍。该时钟节拍同时还是任务调度的时间基准。uC/OS II提供了与时钟节拍相关的系统服务，允许任务延时一定数量的时钟节拍或按时、分、秒、毫秒进行延时。

4. 内存管理

ANSI C中，一般采用malloc()和free()两个函数动态地分配和释放内存。这样，随着内存空间的不断分配与释放，就会把原来很大的一块连续内存区域逐渐地分割成许多非常小的但彼此之间又不相邻的内存块，也就是产生内存碎片问题。为了消除多次分配与释放内存所引起的内存碎片，uC/OS II把连续的大块内存按分区来管理。每个分区中都包含整数个大小相同的内存块，但不同分区之间内存块的大小可以不同。用户需要动态分配内存时，可选择一个适当的分区，按块来进行内存分配；释放内存时，将该块放回它以前所属的分区。这样就能有效解决内存碎片问题，而且每次调用malloc()和free()分配和释放的都是整数倍的固定内存块长，这样执行时间就是确定的了。

4.2.1 uC/OS II 在 ATmega128 上的移植

4.2.1.1 uC/OS II 的可移植性分析

所谓移植，就是使现有的操作系统能在一个特定的处理器上运行。uC/OSII的结构使它有很好的移植性，为了方便移植，大部分uC/OSII代码为使用ANSI C语言写的平台无关性代码，但仍然需要用C语言和汇编语言写一些与处理器相关的代码，这是因为uC/OS II在读写处理器寄存器时只能通过汇编语言来实现。要使uC/OS II正常运行，处理器必须满足以下要求：

- (1) 处理器的C编译器能产生可重入代码；
- (2) 用C语言就可以打开和关闭中断；
- (3) 处理器支持中断，并且能产生定时中断(通常在10Hz至100Hz之间)；
- (4) 处理器支持能够容纳一定量数据(可能是几千字节)的硬件堆栈；
- (5) 处理器有将堆栈指针和其它CPU寄存器读出和存储到堆栈或内存中的指令。

从对目标处理器ATmega128的分析可知，它满足移植uC/OS II所需的各项要求。根据处理器结构的不同，移植的代码量不尽相同，但是代码完成的工作基本上是一致的。uC/OS II的代码分为两大部分：平台相关代码和平台无关代码，其软硬件体系结构如图4.4所示。

从uC/OS II的源代码结构可知，uC/OS II中与硬件相关的代码全部都在OS_CPU.H、OS_CPU_A.ASM和OS_CPU_C.C这三个文件中，因此，移植的相关工作也是在这个三个文件中进行。

OS_CPU.H，定义移植了一些相关的数据类型；

OS_CPU_A.ASM，定义了与任务堆栈建立和任务的上下文切换所进行的操作和与处理器相关的进出临界代码的方法。

OS_CPU.C，定义了七个与操作系统相关的消息钩子函数的用户实现^[39]。

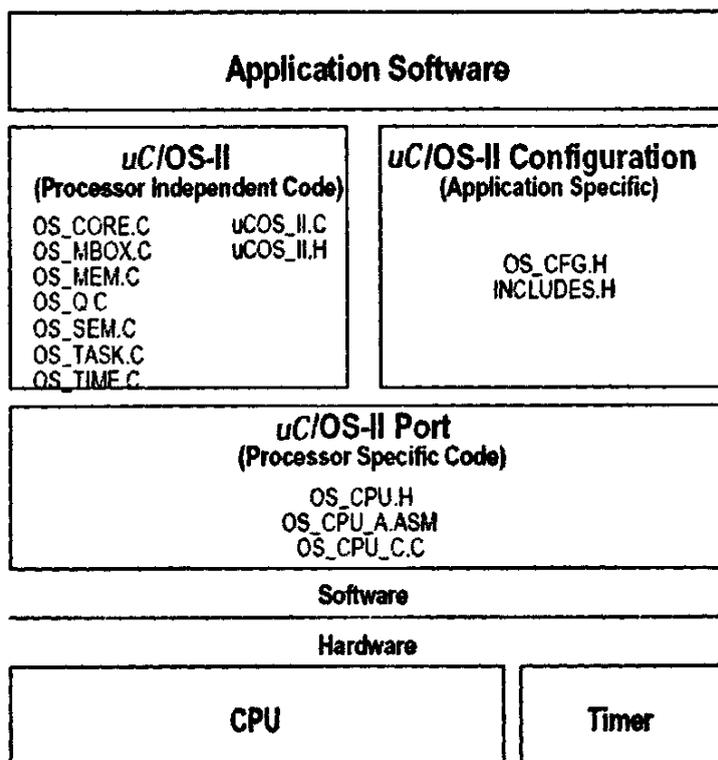


图4.4 uC/OS II 系统框架

4.2.1.2 与编译器相关数据类型

因为不同的微处理器有不同的字长，所以 uC/OS II 的移植包括了一系列的类型定义。uC/OS II 代码从不使用 C 的 short, int 和 long 等数据类型，因为它们是与编译器相关的，不可移植，所以在 uC/OS II 代码中是使用的是一些类型定义来表示这些数据类型。以下是根据 ATmega128 处理器所定义的类型(在 OS_CPU.H 中):

```

typedef unsigned char  BOOLEAN;      //布尔型
typedef unsigned char  INT8U;        //8 位无符号字符
typedef signed char    INT8S;        //8 位有符号字符
typedef unsigned int   INT16U;       //16 位无符号整型
typedef signed int     INT16S;       //16 位有符号整型
typedef unsigned long  INT32U;       //32 位无符号整形
  
```

```

typedef signed long INT32S; //32 位有符号整形
typedef float FP32; //浮点数
typedef unsigned char OS_STK; //堆栈每项 8 位

```

4.2.1.3 进出临界代码段

uC/OS II需要先禁止中断再访问代码的临界段，并且在访问完毕后重新允许中断，这就使得uC/OS II能够保护临界段代码免受多任务或中断服务程序(ISRs)的破坏。因此在uC/OS II定义中有了两个宏操作

OS_ENTER_CRITICAL()和OS_EXIT_CRITICAL()来禁止和允许中断，在程序中执行临界代码段时使用以下方法：

```

{
    OS_ENTER_CRITICAL();
    //临界代码段
    OS_EXIT_CRITICAL()
}

```

移植到 ATmega128 上，需要在 OS_CPU_A.ASM 定义两个宏，如下：

```

OS_ENTER_CRITICAL();
IN    R16,SREG; //保存寄存器值
CLI; //禁止中断
OS_EXIT_CRITICAL();
OUT   R16,SREG; //装载寄存器值

```

4.2.1.4 OSTaskStkInit()

在多任务系统中，每个任务在内存中都对应有一个叫任务控制块(Task Control Block)的数据结构，它能确保任务从当时被中断的那一点丝毫不差地继续运行。在任务控制块中包含有描述任务的各种信息，其中就包括了任务堆栈。当任务正常运行时，任务堆栈就像平常的程序堆栈一样被使用，但当要发生任务切换时，需要把当前各个寄存器的值压入任务堆栈，用来保存此

时任务运行的上下文环境(context)，以便于下次要恢复任务运行时能保证各个寄存器的值能与任务被中断前完全一致。

在创建任务的时候，需要对任务堆栈进行初始化，使堆栈看起来就像刚发生过中断并将所有寄存器保存到堆栈中的情形一样。因为不同的处理器具有不同的寄存器组织方式，所以在移植时需要针对目标处理器编写堆栈初始化函数OSTaskStkInit()，该函数在OS_CPU_C.C中，下面是针对ATmega128编写的堆栈初始化函数所生成的堆栈结构，如图4.5所示^[40]。

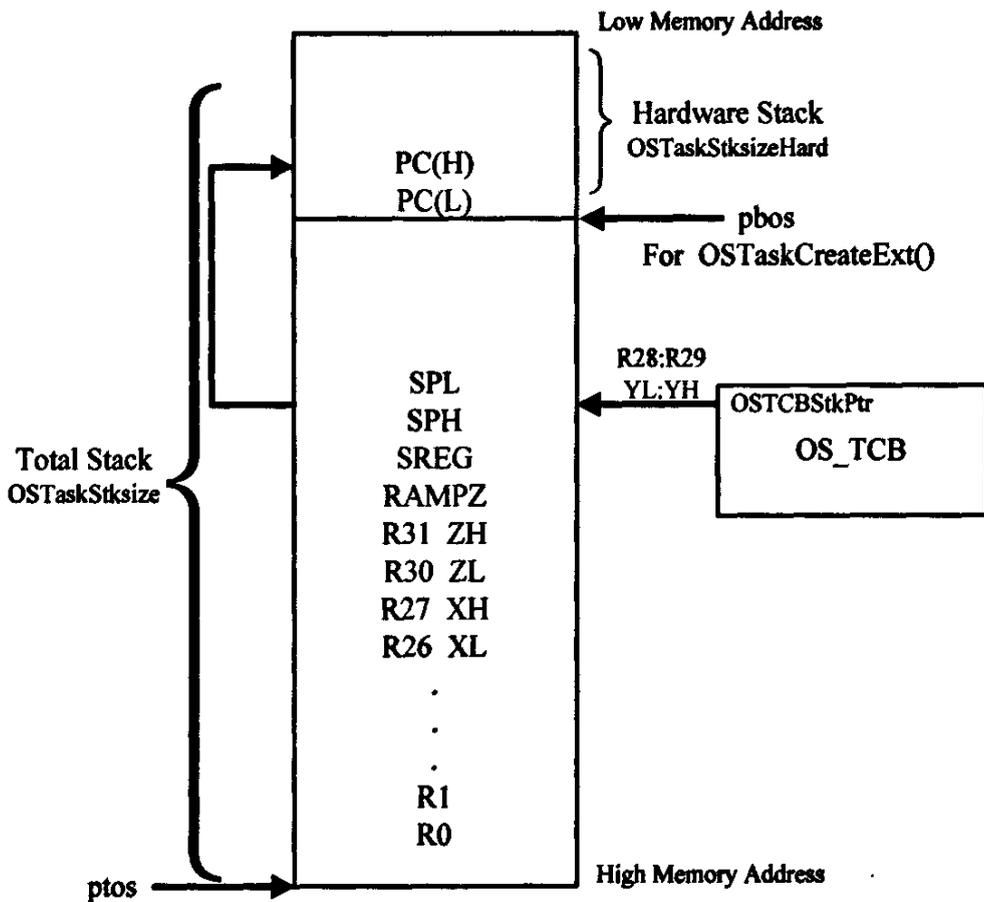


图4.5 ATmega128的任务堆栈

当执行完堆栈初始化函数后，任务堆栈中便保存了一份手动设置的寄存器初始值，当任务被运行时，堆栈中的各个寄存器值将被写入ATmega128中相应的物理寄存器。

4.2.1.5 OSStartHighRdy()

使就绪状态任务开始运行的函数叫 OSStart(), 在调用 OSStart()之前, 用户必须至少已经建立了自己的一个任务打算 OSStartHighRdy()就是用来使优先级最高的就绪任务投入运行。前面在介绍任务堆栈初始化函数的时候已经提到, uC/OSII 中处于就绪状态的任务的堆栈结构看起来就像刚发生过中断并将所有的寄存器保存到堆栈中的情形一样, 要想运行最高优先级的任务, 所需要做的就是将所有处理器寄存器按顺序从任务堆栈中恢复出来, 并且执行中断的返回。为了简化这一点, 堆栈指针总是存储在任务控制块(TCB)的开头, 这就是说, 要想恢复的任务堆栈指针总是存储在 TCB 的 0 偏移内存单元中。它也是一个和底层有关函数, 需要用汇编指令来编写。在 ATmega128 上的移植代码如下:

```
OSStartHighRdy: CALL  OSTaskSwHook          ; Invoke user defined context switch hook
                 LDS   R16,OSRunning        ; Indicate that we are multitasking
                 INC   R16                  ;
                 STS   OSRunning,R16        ;

                 LDS   R30,OSTCBHighRdy    ; Let Z point to TCB of highest priority task
                 LDS   R31,OSTCBHighRdy+1  ; ready to run
                 LD    R28,Z+                ; Load Y (R29:R28) pointer
                 LD    R29,Z+                ;

                 POP_SP                      ; Restore stack pointer
                 POP_SREG_INT                ; Restore status register (DISABLE interrupts)
                 POP_ALL                     ; Restore all registers
                 RETI
```

4.2.1.6 OSCtxSW()

OSCtxSW()是在 uC/OSII 从低优先级任务切换到高优先级任务时被调用的, 而且总是在任务级代码中被调用。任务切换只是简单的将处理器寄存器保存到将被挂起的任务堆栈中, 并且将更高优先级的任务堆栈恢复出来。因为任务切换需要对寄存器操作, 所以 OSCtxSW()需要用汇编代码编写, 下面是针对 ATmega128 移植编写的 OSCtxSW()代码:

```

OSCtxSw:   PUSH_ALL                ; Save current task's context
           PUSH_SREG
           PUSH_SP

           LDS  R30,OSTCBCur        ; Z = OSTCBCur- >OSTCBStkPtr
           LDS  R31,OSTCBCur+1      ;
           ST   Z+,R28              ; Save Y (R29:R28) pointer
           ST   Z+,R29              ;

           CALL OSTaskSwHook        ; Call user defined task switch hook

           LDS  R16,OSPrioHighRdy    ; OSPrioCur = OSPrioHighRdy
           STS  OSPrioCur,R16

           LDS  R30,OSTCBHighRdy     ; Let Z point to TCB of highest priority task
           LDS  R31,OSTCBHighRdy+1   ; ready to run
           STS  OSTCBCur,R30         ; OSTCBCur = OSTCBHighRdy
           STS  OSTCBCur+1,R31      ;

           LD   R28,Z+              ; Restore Y pointer
           LD   R29,Z+              ;

           POP_SP                   ; Restore stack pointer
           LD   R16,Y+              ; Restore status register
           SBRC R16,7               ; Skip next instruction in interrupts DISABLED
           RJMP OScCtxSw_1

           OUT  SREG,R16            ; Interrupts of task to return to are DISABLED
           POP_ALL
           RET

OSCtxSw_1: CBR  R16,BIT07          ; Interrupts of task to return to are ENABLED
           OUT  SREG,R16
           POP_ALL                  ; Restore all registers
           RETI
    
```

4.2.1.7 OSIntCtxSw()

OS_TASK_SW()只能在任务级代码中调用，而不能在中断服务程序(ISR)中调用。这是因为，在程序进入中断服务程序之后，处理器的所有寄存器已经被保存到了堆栈中，并且处理器进入了中断模式(irq)，此时系统所使用的堆栈是中断堆栈而不是任务堆栈。如果要在中断程序中进行任务切换，需要先把各个寄存器的值从中断堆栈转移到被中断任务的任务堆栈，再把新任务切换进来。这就是OSIntCtxSw()的作用，它用于在中断服务程序(ISR)中进行任务切换，同样地，它也要用汇编指令来编写。在ATmega128上的移植代码如下：

```

OSIntCtxSw:
    CALL    OSTaskSwHook          ; Call user defined task switch hook

    LDS    R16,OSPrioHighRdy      ; OSPrioCur = OSPrioHighRdy
    STS    OSPrioCur,R16

    LDS    R30,OSTCBHighRdy      ; Let Z point to TCB of highest priority task
    LDS    R31,OSTCBHighRdy+1    ; ready to run
    STS    OSTCBCur,R30          ; OSTCBCur = OSTCBHighRdy
    STS    OSTCBCur+1,R31        ;

    LD     R28,Z+                 ; Restore Y pointer
    LD     R29,Z+                 ;

    POP_SP                        ; Restore stack pointer
    LD     R16,Y+                 ; Restore status register
    SBRC   R16,7                  ; Skip next instruction in interrupts DISABLED
    RJMP   OSIntCtxSw_1

    OUT    SREG,R16              ; Interrupts of task to return to are DISABLED
    POP_ALL
    RET

OSIntCtxSw_1: CBR    R16,BIT07    ; Interrupts of task to return to are ENABLED
              OUT    SREG,R16
              POP_ALL            ; Restore all registers
              RETI

              END
    
```

4.2.1.8 OSTickISR()

uC/OSII要求提供一个时钟资源来实现时间的延时和期满功能，时钟节拍应该每秒钟发生10~100次。在uC/OS II的代码中有一个函数OSTimeTick()用来处理时钟中断，但是不能够直接把这个中断服务程序的地址直接放到中断向量表上。因为要能在中断服务程序结束之后恢复被中断任务的运行，就必须恢复处理器所有寄存器的值。在ATmega128上选用Timer2定时溢出中断作为时钟节拍中断，一个时钟节拍为10ms。具体在ATmega128上的实现代码如下：

```

BSP_TickISR:
    PUSH_ALL                      ; Save all registers and status register
    IN     R16,SREG                ; Save the SREG but with interrupts enabled
    SBR   R16,BIT07
    ST    -Y,R16
    PUSH_SP                        ; Save the task's hardware stack pointer onto task's stack

    LDS   R16,OSIntNesting        ; Notify uC/OS-II of ISR
    INC   R16                      ;
    STS   OSIntNesting,R16        ;
    
```

```

CPI   R16,1           ; if (OSIntNesting == 1) {
BRNE  BSP_TickISR_1

LDS   R30,OSTCBCur    ; OSTCBCur->OSTCBStkPtr = Y
LDS   R31,OSTCBCur+1
ST    Z+,R28
ST    Z+,R29          ;}

BSP_TickISR_1:
CALL  BSP_TickISR_Handler ; Call tick ISR Handler written in C

CALL  OSIntExit        ; Notify uC/OS-II about end of ISR

POP_SP                 ; Restore the hardware stack pointer from task's stack
POP_SREG_INT
POP_ALL                 ; Restore all registers
RETI

END
    
```

4.2.2 任务的分配与构造

任务的构造和分配的思路是：将整个系统划分成 10 个任务。其系统结构如图 4.6 所示。

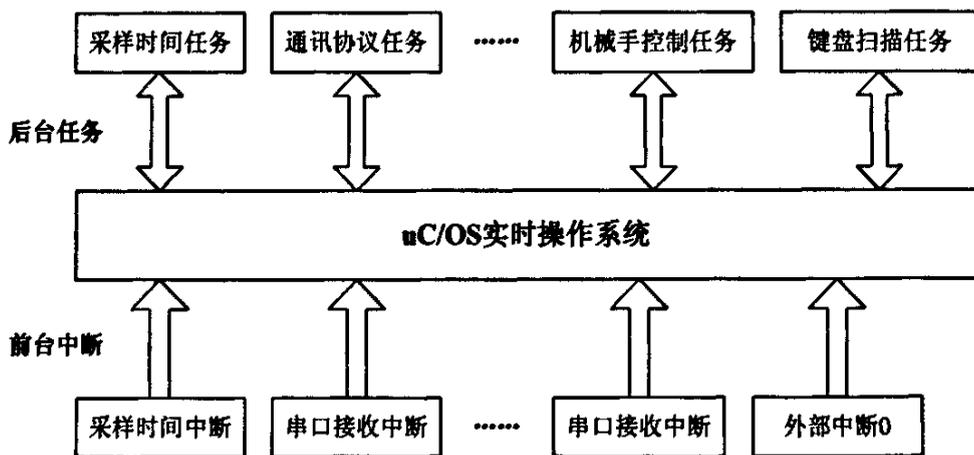


图 4.6 系统任务结构图

(1) 初始化任务，初始化整个硬件系统，初始化时钟节拍中断；建立其他任务，完成一切初始化任务后降低其自身的任务优先级，作为后台任务，1s 运行一次。

(2) 串口接收任务，当串口中断接收到数据时，调用 OSSemPost() 释放信号量 OSUartSem，串口接收任务由等待状态进入就绪状态，中断退出后，

若此时系统中没有比串口接收任务高优先级的任务，则此任务进入运行状态。此任务判断收到的数据内容是否是有效帧，若数据为有效帧，再获取帧数据后，任务调用 `OSMboxPost()` 以消息邮箱（`UatrReviceMbox`）的方式，发送给协议解析任务。完成后，任务再次调用 `OSSemPend()` 等待 `OSUartSem` 信号量，若无信号量，则任务进入等待状态。

(3) 通讯协议解析任务，当任务接收到邮箱 `UatrReviceMbox` 后，解析帧的内容，并根据系统当时的状态来响应操作，然后任务调用 `OSMboxPend()` 再次等待邮箱 `UatrReviceMbox`，若没有有效帧时，则任务进入等待状态。

通讯任务之间通讯及转换关系如图 4.7 所示。

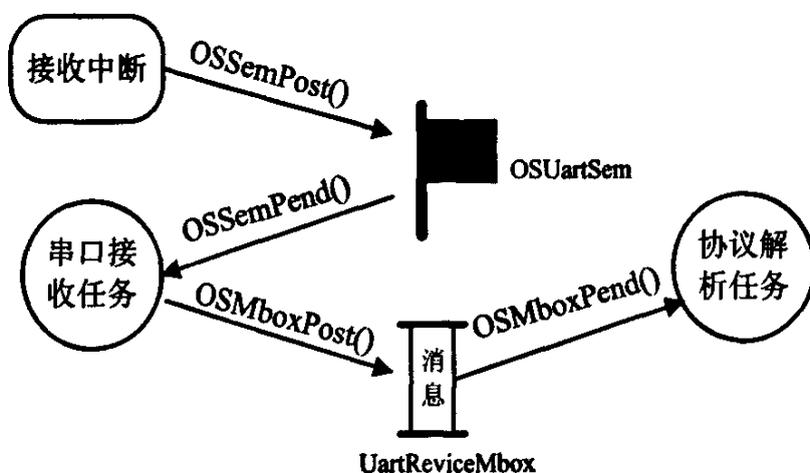


图 4.7 通讯任务调度关系

(4) 行程开关扫描任务，此任务对时间不敏感，通过调用 `OSTimeDly()`，使该任务每 10ms 执行一次。在任务中，对各个关节的按键进行循环扫描，检测到按键后，进行延时、去抖。当确定有键按下后，设置键值和有键按下的标志。

(5) 复位任务，等待来自协议解析任务的复位命令信号量。运行时，机器人各个关节以闭环的方式，向一边的行程开关靠近，实现对各个关节的相对编码器进行标定。

(6) 采样时间任务，是对时间非常敏感的任务，影响整个控制系统的精度和响应，因此利用 `ATmega128` 内部的定时器 `Timer0` 产生比较中断，触发

5ms 采样时间的定时信号量 OSTimeSem，以准确的触发采样时间任务。为了保证该任务的执行时间，设置此任务的优先级为所有任务中最高的。采样时间任务里对各个关节进行位置和速度闭环伺服计算，输出控制量；图 4.8 为采样时间任务调度关系。

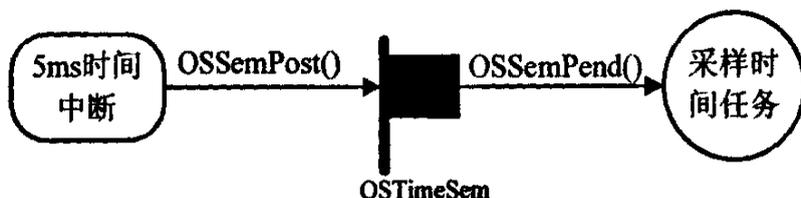


图 4.8 采样时间任务的调度

(7) AD 采样任务，借助消息队列，其他任何任务通过调用 OSQPost() 向消息队列发送消息，以终止 AD 采样任务的等待状态，使其马上执行一次 AD 采样。消息的内容包括：AD 通道、采样频率。得到结果后 AD 采样任务调用 OSMboxPost() 通过邮箱将 AD 采样的结果返回给任务，图 4.9 为 AD 采样任务的调度关系。

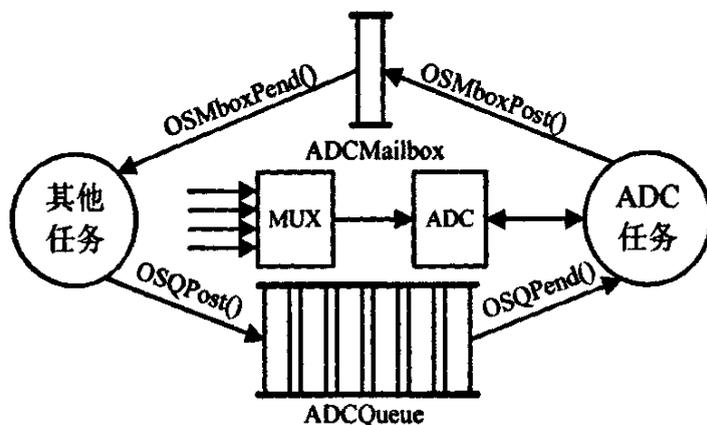


图 4.9 AD 采样任务的调度

(8) 手爪控制任务，等待协议解析任务发出的信号量，进入运行状态后，根据命令以及当前手爪的行程开关的位置控制手爪的开合。

(9) 系统空闲任务，当所有的任务进入等待状态后，运行系统空闲任务。

(10) 系统统计任务，系统统计任务，每秒运行一次，得到 CPU 的使用

率。

各个任务之间固定优先级，各自的堆栈空间，其中系统采样时间任务的优先级最高，以保证系统的控制性能。实时操作系统 uC/OS II 构架的建立，在硬件速度、程序空间和数据空间允许的前提下，大大改善了系统的实时性、可扩展性、可靠性、可维护性。

4.2.3 性能的总体评价

机器人包括腰回转、垂直臂俯仰、水平臂俯仰、腕回转、腕俯仰五个自由度和一个手爪的张/合的控制，所示机器人的实物照片如图 4.7 所示。主要参数如下：负载能力 1kg，关节定位精度为 0.01 度，采用 FAULHABER 公司的 2342S024CR 型直流伺服电动机，24V/30W；编码器与电动机集成于一体，每转脉冲数为 512；功率驱动元件为 IRF540，100V/150w^[4]。



图 4.7 机器人照片

采用数字 PID 控制算法实现五个关节和手爪的位置、速度双闭环伺服控制，系统采样周期为 5ms。实验结果表明，系统可以在 5ms 采样周期条件下实现机器人五个伺服回路的位置伺服控制，驱动电路工作正常，控制系统运行稳定，抗干扰能力好。各个关节均无超调，定位精度为正负 1/4 个编码器脉冲（4 倍频编码器接口）。由于各个关节的传动比是不同的，关节实际定位

精度也是不同的。以腕回转关节为例，其减速比为 246，关节的实际定位精度为 0.0014 度。由于各个关节的驱动电机是相同的，而负载情况是不同的，其响应速度也有所差别，其中腕回转关节响应速度最高，带宽为 20Hz，其余关节的带宽在 10~20Hz 之间。而且算法经过优化后，在统计任务中，得到 CPU 的使用率仅 29%，说明伺服控制器能进行更加复杂的算法，有很大的潜力。

综上所述，所设计的六自由度机器人直流伺服控制器以 ATmega128 为运算核心，FPGA 为外围扩展的硬件构架，充分的利用了 ATmega128 的强大高速的计算能力。FPGA 的外围扩展不仅提高了系统的集成度，而且提升的系统的灵活性。两者的结合为高速，高精度的多自由度的运动控制提供了很好的数字控制平台。实时操作系统 uC/OS II 的引入，在不降低实时性的情况下，加快了软件开发的进度，提高了软件的可靠性，可维护性。系统采样周期可达 5ms，可以满足 20Hz 带宽的多自由度机器人控制要求。驱动能力的大小主要取决于大功率场效应驱动元件本身的特性，选择不同的场效应驱动器件可以获得不同的驱动能力，文中应用实例的实际驱动能力可以达到 200w。所设计的六自由度机器人伺服控制系统不但可以用于六自由度机器人的控制，也可以推广应用到其他多自由度直流伺服系统^[42, 43]。

4.3 通讯协议的设计

五自由度教学机器人采用 RS232 协议与上位机进行通讯。上位机的软件采用 VC++6.0 SP6 编程，通过对 MSCOMM 控件编程和下位机进行通讯。

4.3.1 通讯协议的要求

根据第二章的分级控制的总体方案，下位机实现位置、速度伺服，上位机实现机器人运动学的逆解。下位机相对于上位机是一个黑盒，并提供唯一的位置输入的接口，因此上下位机之间需要对各个关节的当前的位置数据和手爪的开合状态做数据交换，在通讯的过程中，下位机始终属于被动状态，即下位机只接收上位机查询，不主动向上位机发送数据^[44]。

将上下位机的数据交换指令分为两类：

(1)数据交换指令,主要指教学机器人的各个关节的位置数据交换指令,包括上位机的位置给定及下位机的位置反馈。

(2)命令指令,主要是指上位机对下位机有特殊的控制指令,如复位,握手等。

当下位机检测到机器人处于错误状态时,需要向上位机返回错误代码,以供上位机对错误状态做出及时的处理。

4.3.2 通讯协议的建立

根据上述的要求建立符合要求的通讯协议,采用两种长度的数据帧的形式,一种单字节的命令帧,另一种为18字节的数据交换帧,为了防止命令字与数据的干扰,命令字采用大于0xAA的十六进制数,数据采用0~9的压缩BCD编码的方式,具体的协议如下:

1. 数据帧形式

基本形式:1个字节命令字+16个字节数据+1个字节校验;

简述:命令字为十六进制大于0xAA的一个字节,16个数据分别代表6个自由度的关节角度值或编码器数。其中5个自由度的表示用如下形式:

给编码器数时: 55 | 55 | 55 | 范围:0—999999

一字节 一字节 一字节

给角度时: 01 | 80. | 00 | 范围:0.00—180.00度

一字节 一字节 一字节

小数点

注:数据的表示形式都为4位的压缩BCD码。即:0x99表示十进制数99;因此以上五个自由度表示用15字节。

手爪自由度:手爪自由度控制为一个字节开关量。即1—张开,0—闭合。

五自由度加手爪自由度在数据帧中的排列:

手爪自由度+第一自由度+第二自由度+第三自由度+第四自由度+第五自由度;

校验方法：16 个数据的异或和。

2. 数据通信物理规范

方式：RS232，三线通信方式

规范：波特率 115200bit/s，通信 10 位方式，1 位起始，8 位数据，1 位停止，无奇偶校验位。

3. 命令说明

0xEA 握手命令。在下位机就位后等待主机命令，在此之前其他命令无效。在结束命令后，须重新握手。

从机返回：0xEA

0xEB 复位命令。在握手以后各关节进入复位状态。各关节分别向左右极限复位。

从机返回：EB+一个字节的复位状态字

复位状态字简述：

0xEB 1 1 1 1 1 1 1 1

保留 保留 手爪 第一第二第三第四第五

1: 成功 0: 失败

0xDA+数据+校验码

位置给定命令。数据值位各个关节的编码器值，手爪为开关控制。数据格式如上述。

从机返回：从机返回：校验正确返回 0xAD，错误返回 0xEE

0xAC 脉冲方式给定命令。收到该命令从机进入脉冲给定方式

从机返回：无

0xAD 主机查询位置返回命令。

从机返回：0xAD+数据+校验位
格式如上所述，返回角度值。

0xBD 主机查询位置返回命令。
从机返回：**0xBD+数据+校验位**
格式如上所述，返回编码器值。

0xDB+数据+校验码

位置给定命令。数据值为角度值，手爪为开关控制。数据格式如上所述。

从机返回：无

0xEE 下位机停止命令。停机后，要重新握手才有效。
从机返回：**0xEE**

各个关节左右行程到返回格式：

0xEA 关节号 表示左限位到

0xEB 关节号 表示右限位到

4.4 本章小结

本章详细的阐述了五自由度教学机器人的伺服控制器的软件系统构成，包括数字 PID 控制的软件的设计、uC/OS II 实时操作系统在 ATmega128 上的移植、在 uC/OS II 平台上开发了伺服控制软件，设计了一套面向多关节控制的通讯协议。

第 5 章 样机的研制及实验研究

五自由度的教学机器人的实验研究主要包括：FPGA 的调试、单片机的调试、uC/OS II 的软件调试、控制系统的性能调试和机器人示教再现实验^[46, 48]。

5.1 控制系统软硬件调试

本教学机器人的控制系统硬件组成如第三章所述，主要由两大部分组成，即 FPGA 电机控制外设模块和主单片机控制系统。其中 FPGA 电机控制外设模块的调试是利用 ALTERA 公司的 Quartus II 软件电路仿真验证和 ByteBlaster 下载电缆完成该模块的调试的；主单片机控制系统则是利用 AVR 单片机系统内集成的 JTAG（边界扫描协议）调试接口，通过仿真器完成电机控制实验的调试的。

软件调试主要是对 uC/OS II 实时操作系统的调试，利用了 IAR Embedded Workbench for Atmel AVR V4 Evaluation 中内嵌的调试插件：uC/OS II Plugins 完成的。

5.1.1 FPGA 的调试

FPGA 的开发流程主要包括：软件要求、编程实现、软件仿真和下载验证。

目前 FPGA（现场可编程逻辑门阵列）编程方法主要包括三种：原理图方式编程、HDL（硬件描述语言）编程和两者混合编程，图形方式编程利用编译器提供的成型上手容易，适用于规模小的系统，比较直观，但对编译器的依赖严重，可移植差；HDL 的编程方式遵循具体的标准利用具体的语言描述组合电路和时序电路，其编程思想和传统的编程思想有一定的差异，不依赖具体的编译器和 FPGA 芯片，具有很高的可移植性，但对编程人员有一定的要求；混合编程的方法主要应用在大规模的系统中，用 HDL 语言对各个模块

进行描述，然后在顶层用图形化参数的方式进行组合，有利于系统的维护。

软件仿真通过专业的软件，对编程所描述的电路进行描述，验证电路的输出是否能满足要求。目前进行仿真的方法有两种：图形化仿真和 HDL 语言实现。图形化仿真的方法，根据要求用图形的方式描述输入波形进行仿真，直观但速度较慢；HDL 语言则通过语言编程描述实现输入波形，比较精确多应用在专业的领域。

硬件描述语言 HDL(Hardware Description Language)是电子系统硬件行为描述、结构描述、数据流描述的语言。数字电路设计者利用这种语言可以从上层到下层，逐渐描述自己的设计思想，用一系列分层次的模块来表示极其复杂的数字系统。比起传统的原理图设计方法，HDL 有许多优点，主要有：

(1) 用 HDL 设计电路能够获得非常抽象的描述。设计者不用选择特定的制造工艺就能写出电路的寄存器传输级(Register Transfer Level, RTL)描述。逻辑综合工具能自动将 RTL 描述转换成任何一种制造工艺。

(2) 用 HDL 描述电路设计，在设计的前期就可以完成电路功能级的验证。

(3) 用 HDL 设计电路类似于计算机编程。

硬件描述语言 HDL 是本次开发过程中不可缺少的工具。

本系统的 FPGA 电机控制外设模块利用了 ALTERA 公司提供的 Quartus II 集成软件建立系统。Quartus II 是 ALTERA 推出的新型的 FPGA 开发软件，支持所有 ALTERA 的 FPGA 器件和 CPLD(可编程逻辑器件)器件，集成了编程、布局、仿真和下载所有功能的集成开发环境。采用 Verilog HDL 对电路进行编程，利用 Quartus II 本身提供的图形仿真工具对模块进行仿真验证，最后对系统进行时钟逻辑评估，以得出模块的最高的工作频率，经过此软件的评估，本电机控制外设模块的最高频率为 49.02MHz，满足要求。

在进行 FPGA 软件设计的过程中，在设计与 ATmega128 地址/数据总线进行数据交换模块中，遇到了比较大的困难。由于 ATmega128 是 8 位的高速 CPU，在地址/数据总线的速度非常快，而 FPGA 内的寄存器为 16 位，因此当发生数据访问时，数据锁存操作，当 FPGA 采用 20MHz 的时钟时，根本无法满足地址/数据总线的速度。经过反复实验和仿真，通过减少数据访问中不必要的状态等待和提高 FPGA 系统时钟频率，解决了困难，并在实际的实

验中反复的检测模块的正确性，验证了模块的可靠性。

5.1.2 单片机总体硬件的调试

整个伺服控制的硬件系统采用+24V 单电源供电，分为控制和驱动电路两部分。控制电路的电源和驱动电源隔离分开供电，采用光耦来传递信号 [47, 48]。

控制电路部分包括：6 路正交编码器接口、6 路脉冲计数接口、一个 16 位可编程 I/O、一个 8 位可编程 I/O、6 路独立的 PWM 输出、32K 的 SRAM、1Mbit 的 Flash 的数据存储空间、2 个 RS232 的串口、4 路模拟量输入接口，由于 FPGA 可以通过重新编程实现其他外围电路的功能，因此控制部分有很强的可扩展性，方便以后的二次开发和使用，主控电路板如图 5.1 所示。

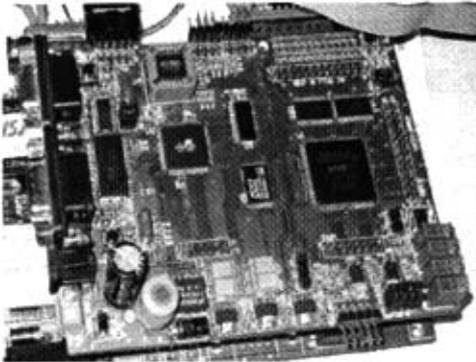


图 5.1 主控电路板

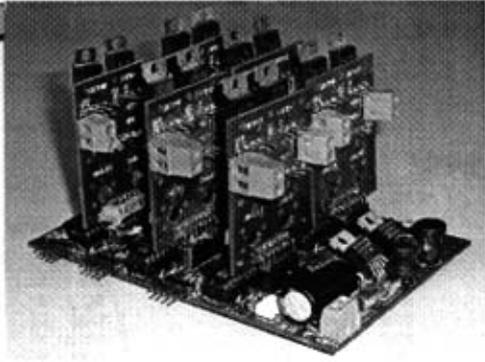


图 5.2 驱动电路板

驱动电路包括：电源隔离模块、6 路功率放大模块、6 路电流保护电路，驱动板如图 5.2 所示

当完成对 FPGA 的程序编制和调试后，FPGA 模块对于单片机来说就是一个黑箱，单片机以外设的方式访问此模块。在调试的过程中，采用模块化调试的方法，先对单独的某个模块进行编程测试，然后进行集成，这样能避免调试过程中出现的不必要的耦合现象，容易确定问题的所在。

在调试单片机的调试的过程中，首先要对 ATmega128 的熔丝位进行编程，因为每片 ATmega128 的出厂设置均为兼容 ATmega103 模式，此模式下

地址总线的模式和标准的不同，而且使用的片内的晶振，片内的晶振频率不是很准确，这将直接导致串口通信波特率不准，造成通讯错误，忘记配置熔丝位是一个比较容易忽视的问题；再者当发现问题后，先检查复位，笔者好几次就是因为复位的问题，导致程序不正常。

5.1.3 uC/OS II 的软件调试

整个系统的软件编程基本使用 C 语言实现，采用 IAR Embedded Workbench 的编译器。IAR 是国际知名的编译器软件开发商，在代码生成的大小和效率方面都有很高的造诣，C 语言的编译效率在 90%以上，而且在调试方面，集成了大量的调试工具，如数据覆盖、代码覆盖、堆栈剖析和实时操作系统支持。

为了使用这个插件，需要在 uC/OS II 的 OS_CFG.H 中设置 OS_DEBUG_EN 为 1，然后在编译器的编译选项里打开控件，并重新编译、下载，进入 uC/OSII 调试界面，如图 5.3。

Name	Ref	Prio	State	Dly	Waiting On	Msg	Ctx Sw	Stk Ptr	Mem%	Cur%	Mem	Cur	Size	Starts	Ends
SampleTimeTask	3	1	Sem	0	?		6172	89A7	0	0	0	0	0000		
KeyScanTask	4	3	Dly	1			6173	838C	0	0	0	0	0000		
KeyDoTask	5	4	Mbox	0	?		5967	84A1	0	0	0	0	0000		
Uart1RcvcsTask	8	5	Sem	0	?		1	8BC4	0	0	0	0	0000		
CommTask	10	6	Mbox	0	?		1	8DAD	0	0	0	0	0000		
ResetTask	6	7	O	0	?		2	87BE	0	0	0	0	0000		
HandControlTask	7	8	Mbox	0	?		1	81DB	0	0	0	0	0000		
AD_Task	12	9	O	0	?		17	80A3	0	0	0	0	0000		
Uart1RcvcsTask	9	10	Sem	0	?		1	8CB3	0	0	0	0	0000		
UartDebugTask	11	11	Mbox	0	?		1	8E76	0	0	0	0	0000		
Start Task	2	12	Dly	244			64	8677	0	0	0	0	0000		
uC/OS-II Stal	1	16	Dly	4			617	9165	0	0	0	0	0000		
> uC/OS-II Idle	0	17	Ready	0			12315	9272	0	0	0	0	0000		

图 5.3 uC/OS II 任务调试界面

控件通过仿真器，对 uC/OS II 的各个系统变量进行访问，通过界面，可以清晰看到当前运行的任务、各个任务的状态、延迟、等待的原因、已经切换的次数、堆栈指针等必要的参数，同样也可以看到关于旗语、消息队列、邮箱、信号量等任务之间通讯方式的状态。

同样利用控件和 OS_CFG.H 文件的设置，在编译过程中得到对这个软件系统的总体评价。将 OS_CFG.H 中的 OS_TASK_STAT_EN 设置为 1，即允许

系统的资源统计任务(系统统计任务每 1s 调用一次, 统计在 1s 内, 系统空闲任务的切换次数, 以得出 CPU 当前的使用率), 重新编译, 下载调试后, 将系统运行一段时间后, 得到 CPU 的使用率如图 5.4 所示。

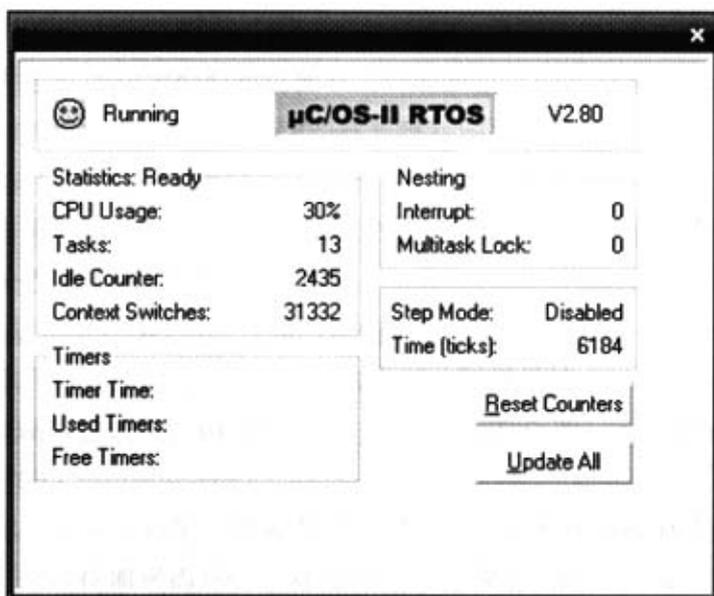


图 5.4 uC/OS II 系统参数界面

图 5.4 中, 当前系统运行的一共有 13 个任务、CPU 的占用率为 30%、已经经过了 6184 个时钟节拍。在完成系统调试后, 可以将调试用的参数删除, 以减少系统的编译后的大小及运行的稳定性。

利用 IAR Embedded Workbench 集成的 uC/OS II 调试插件, 在调试能实时的监控系统的状态和系统的负载状态, 这对加快实时操作系统的软件开发、充分的合理的利用系统的硬件资源, 都有重要意义。

5.1.4 控制系统的性能调试

控制系统的性能参数决定了, 控制系统性能的好坏。数字PID调节器参数的整定可以仿照模拟PID调节器参数整定的各种方法, 根据工艺对控制性能的要求, 决定调节器的参数。这里就各个参数对系统性能的影响简单加以说明:

(1) 比例系数P对系统性能的影响

比例系数加大,使系统的动作灵敏,速度加快,稳态误差减小,P偏大,振荡次数加多,调节时间加长;P太大时,系统会趋于不稳定;P太小,又会使系统的动作缓慢。P可以选负数,这主要是由执行机构、传感器以及控制对象的特性决定的。如果P的符号选择不当对象测量值就会离控制目标的设定值越来越远,如果出现这样的情况P的符号就一定要取反。

(2) 积分控制I对系统性能的影响

积分作用使系统的稳定性下降,I小(积分作用强)会使系统不稳定,但能消除稳态误差,提高系统的控制精度。

(3) 微分控制D对系统性能的影响

微分作用可以改善动态特性,D偏大时,超调量较大,调节时间较短;D偏小时,超调量也较大,调节时间也较长;只有D合适,才能使超调量较小,减短调节时间。

对于本五自由度教学机器人系统,对于五个关节伺服控制方式为位置伺服做外环,速度伺服做内环。这样即保证了位置的精度,又保证了运动速度过程中速度的平稳性。在调试的过程中,对于这种双闭环的控制系统,应该采用先内后外的策略,即先调试速度环,再调试位置环。五个关节的伺服控制相互独立,而且各个关节驱动方式、负载不同,因此调试时,按照自下而上的策略对每个关节的参数进行逐一的调整。

单关节的参数调试采用的硬件在回路的调试方式。通过串口建立PC机和伺服控制器之间数据连接,PC端的软件通过串口得到当前关节的速度和位置的数据,并且用折线图的形式表现出来,如图5.5所示。上位机在线调试控制器的各项参数,然后通过串口下载到主控单片机中,具体结构形式如图5.6所示。

PC端软件主要采用VC++ 6.0 SP6进行编程。在程序里主要使用了两个控件:MSCOMM串口通信控件和MSCHART图表绘制控件。用串口通信MSCOMM控件实现控制PC机串口与伺服系统进行通讯;用MSCHART控件对位置反馈的数据进行实时绘图显示,绘图的采用数据量2000个点,即对系统最近接收到的2000个数据进行绘图。由于Windows不是实时操作系统,系统无法响应伺服控制器每个采样周期的数据,导致死机^[49,50]。

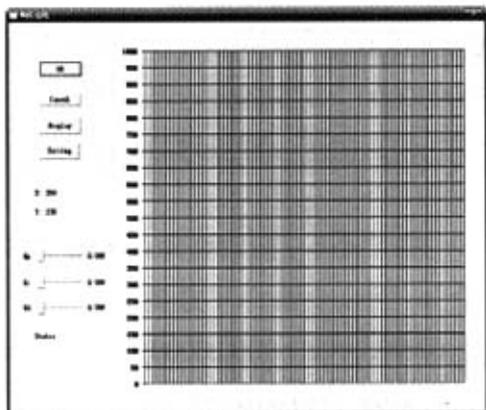


图5.5 软件界面

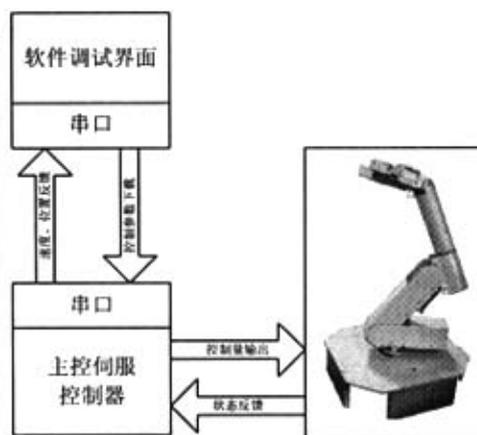


图5.6 单关节参数调试软件结构

解决的方法是将几个采样周期的数据打包，增加每次通讯的数据量，以减少OnComm事件的触发次数，以减少MSCHART控件进行刷新的窗口重绘，从而加快软件系统的响应。采用这个方法在实验中收到了良好的效果。

在伺服控制器的软件的系统中，增加串口调试任务，并对采样时间任务的程序作一定调整：

在采样时间任务增加对每个采样时间节拍时当前位置的存储，当累计到一定的个数（本系统取得是50个采样时间节拍），以消息邮箱的方式，通知串口调试任务。

串口调试任务，解析串口调试协议，控制关节调试的启停，参数的下载及位置反馈数据包的发送。

串口调试协议：

0xBA 调试开始命令。收到该命令下位机机进入电机调试模式

从机返回：无

0xBB 调试结束命令。收到该命令下位机机结束电机调试模式

从机返回：无

0xAA+100 个字节

从机向上位机反馈最近 25 个采样时间的位置数据，每个位置数

据采用 BCD 编码，表示形式如下：

00|00|00|00

表示数的范围：00000000—99999999

从机返回：无

0xAB+参数标识+4 个字节

参数下载命令。用 4 个字节的定点数，表示参数。参数范围 (0000.0000—9999.9999)

参数标识 0x01: K_p 、0x02: K_i 、0x03: K_d

从机返回：无

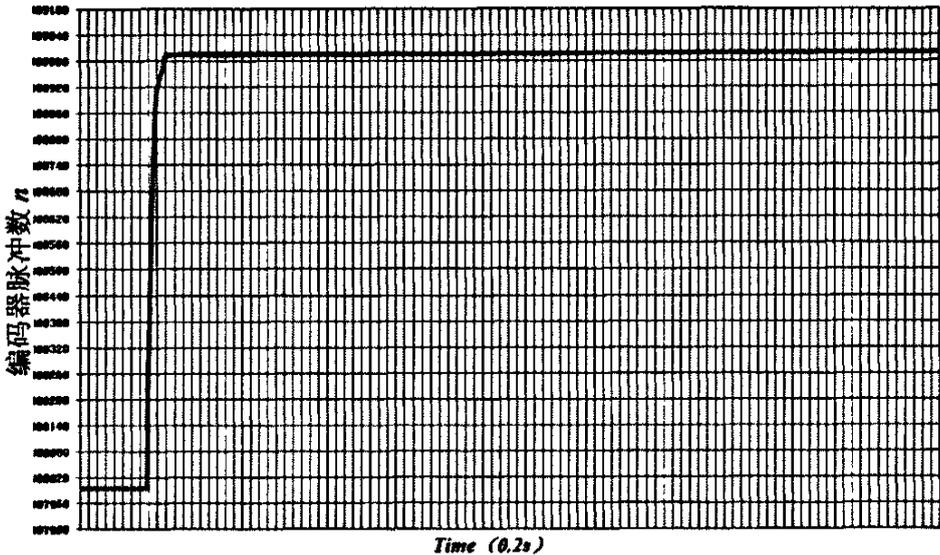


图5.7 单关节的位置响应曲线

图5.7中是五自由度教学机器人的底部回转关节的位置响应曲线，其中位置闭环采用比例调节， $K_p=1$ ；速度内环采用比例积分调节 $K_p=0.25$ ， $K_i=0.001$ ，当前的位置给定为1000个编码器脉冲的位置增量，给定速度为39rad/s，响应时间为0.4s，没有超调，稳态无静差。

5.2 五自由度教学机器人示教再现实验

能够实时的对机器人在线示教，并根据选择不同的记录文件来选择不同的示教程序，让机器人再现不同的示教过程，以达到教学目的为本课题的最终目标。因此，能否实现上述目标就得通过示教再现演示实验来检验。此处只是为了说明问题，因此选择了一组比较简单的示教再现程序，具体操作结果如下：

系统上电自检后，状态如图5.8所示。然后按照操作步骤，对机器人的各个关节进行示教，并将示教点存入PC端的文件里。复位后，选择所示教的数据文件，再现中间状态如图5.9所示。建议示教过程尽量细致，再现的结果就会更好。

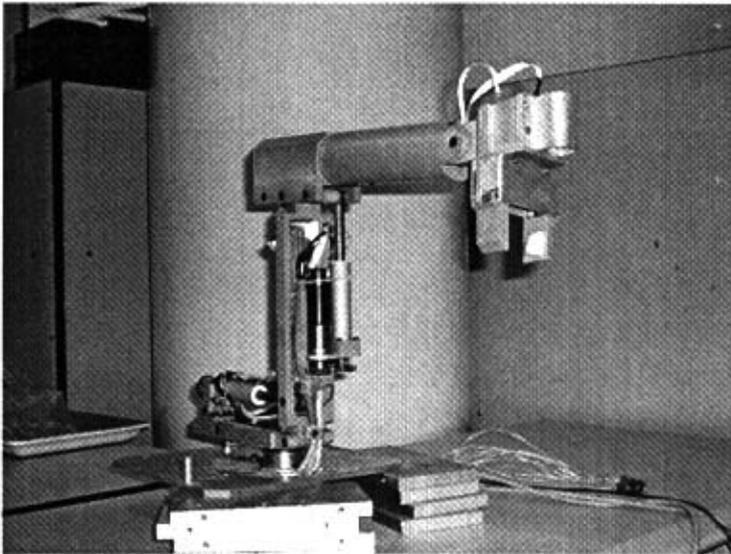
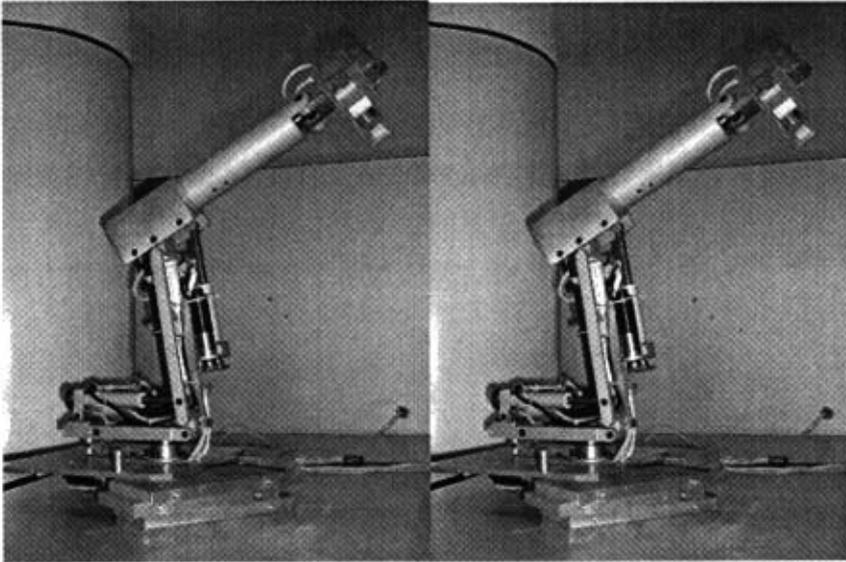
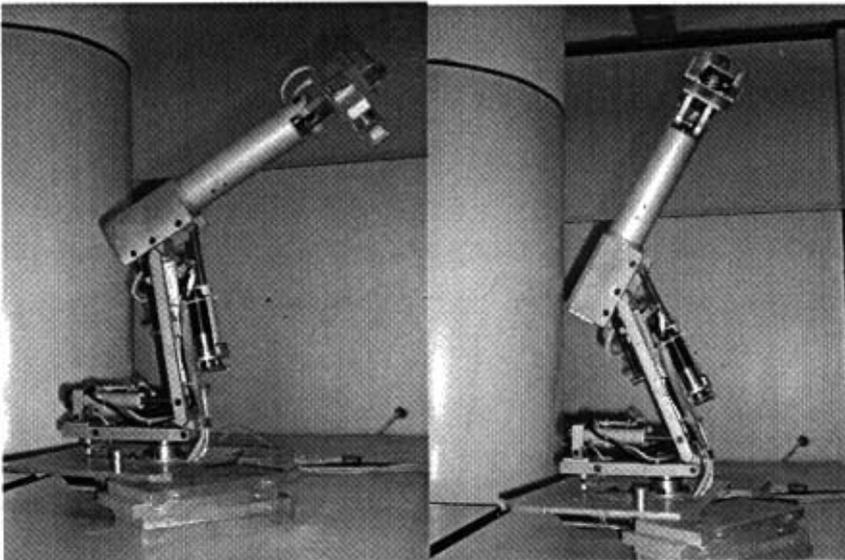


图5.8 机器人上电复位状态



(a)

(b)



(c)

(d)

图5.9 机器人示教再现中间状态

5.3 本章小结

本章详细的阐述了在样机研制过程中所遇到的问题和困难，介绍了本文解决问题的一种方法。主要包括：uC/OS II 的实时操作系统的软件调试；FPGA 的硬件调试；通过串口对单关节的控制系统性能进行调试；通过 PC 端软件对五自由度教学机器人进行了示教再现实验。

结 论

本文结合当今工业机器人的研究现状和发展方向，具体阐述了一种五自由度教学机器人的伺服控制系统的设计与开发过程。实验证明本课题所研制的机器人控制系统可以很好的控制机器人，对机器人多个关节进行高精度的伺服控制，配合 PC 端伺服控制软件，进行示教再现功能。论文主要完成了以下几个方面的工作：

1. 研制了五自由度教学机器人的样机，并进行了示教再现的实验；
2. 设计了多自由度伺服控制器，实现了对多自由度的高精度直流伺服控制；
3. 设计了基于地址总线的 FPGA 电机伺服控制模块，解决了在电机伺服控制中正交编码器的接口难题；
4. 设计了基于 uC/OS II 嵌入式操作系统的，伺服控制系统软件，提高了软件可移植性，可重用性和可靠性，建立了基于串口的单关节伺服控制性能调试平台，实现了对在线的调整伺服控制系统参数；
5. 对各个关节进行了伺服控制实验，实验结果表明，系统可以在 5ms 采样周期条件下实现机器人六个伺服回路的位置伺服控制，驱动电路工作正常，控制系统运行稳定，抗干扰能力好。各个关节均无超调，定位精度为 0.01 度。

本文尚有完善和应进一步开展的研究如下：

1. 实验中发现五自由度教学机器人的传动机构的间隙比较大，特别是进行直角传动的锥齿轮传动精度，影响控制的精度，应通过高精度传动机构来解决。
2. ATmega128 作为主控单片机的资源利用为 30%，还有比较的资源富余，可以进行更加复杂的计算，运用更好的控制算法来提高控制效果。
3. 随着 SOPC(片上系统)的到来，可以运用此技术来为伺服控制建立专门的控制 IC，进一步提高集成度，提高速度及可靠性。

参考文献

- [1] 刘进长, 辛健成. 机器人世界. 河南科学技术出版社, 2000: 117-119页
- [2] 诸静. 机器人与控制技术. 浙江大学出版社, 1991: 7-10页, 115-126页, 127-143页, 222-253页
- [3] 吴广玉, 姜复兴. 机器人工程导论. 哈尔滨工业大学出版社, 1988: 9-17页, 266-278页, 196-241页
- [4] A.C.Mcdonald.Robot Technology:Theory,Design and Appliction. Prentic-Hall, 1986:45-101P
- [5] 周远清, 张再兴. 智能机器人系统. 清华大学出版社, 1989
- [6] 金茂筹, 曲忠萍, 张桂华. 国外工业机器人发展态势分析. 机器人技术与应用. 2001(2): 6-7页
- [7] Jan Karlsson.ROBOT INVESTMENTS SURGE TO RECORD LEVELS.United Nations Economic Commission for Europe. Press Release. Geneva. February 2005: 51-102P
- [8] UNECE issues its 2004 World Robotics survey.United Nations Economic Commission for Europe. Press Release. Geneva,20 October 2004
- [9] 哈尔滨工业大学博实公司主页<http://www.bsjm.cn>
- [10] 上海交通大学机器人研究所网<http://www.robot.sjtu.edu.cn>
- [11] 蔡自兴. 机器人学. 北京: 清华大学出版社, 2000
- [12] 熊有伦. 机器人技术基础. 华中理工大学出版社, 1996.8
- [13] J.J.E.Slotine and W.Li On the adaptive control of robot manipulators.Int.J.Robotics Research,1987,6(3):49-59P
- [14] 张立勋. 机电一体化系统设计. 哈尔滨: 哈尔滨工程大学出版社, 2004
- [15] 张雄伟, 陈亮, 徐光辉. DSP芯片的原理与开发应用(第三版). 北京: 电子工业出版社, 2002
- [16] ATmega128 Datesheet.Atmel,2004.

- [17] 陈伯时, 电力拖动自动控制系统. 北京: 机械工业出版社, 2003
- [18] 张立勋, 王立权, 杨勇. 机械电子学. 哈尔滨工程大学出版社, 1999: 69-95页
- [19] 童诗白, 华成英. 模拟电子技术基础(第三版). 北京: 高等教育出版社, 2002.345-408页
- [20] Altera Corporation.ACEX 1K Programmable Logic Device Family DataSheet.2003
- [21] 阎石. 数字电子技术基础(第四版). 北京: 高等教育出版社, 2000. 132-175页
- [22] 王智慧, 袁梅, 基于DSP和FPGA的通用型伺服控制器设计. 测控技术, 2005(11)
- [23] 邱云周, 张春雷, 高山. DSP与FPGA在运动控制中的应用和研究. 自动化信息, 2003(6)
- [24] 夏宇闻. Verilog 数字系统设计教程. 北京航空航天大学出版社. 2004:1-17页
- [25] 赵灿, 林华促, 田苗淼, 姚远. 基于ARM的多轴伺服控制器设计及其应用. 制造技术与机床. 2006(4)
- [26] H.P.Huang,C.Y.Chiang.DSP-Based Controller for a Multi Degree Prosthetic Hand.IEEE International Conference on Robotics & Automation,2000:1378P
- [27] X.H.Gao.M.H.Jin,The HIT/DIR dexterous hand work progress, IEEE Conf.On Robotics and Automtion,2003,3-164P
- [28] Linght CM,Chappell PH. Development of a lightweight and adaptable multi 2axis hand prosthesis. Medical Engineering and Physics,2000,22(10):66-684P
- [29] 何立民. 单片机应用系统设计与接口技术. 北京航空航天大学出版社, 1990: 407-412页
- [30] 丁化成, 耿德根, 李君凯. AVR单片机应用设计. 北京航空航天大学出版社, 2002: 100-160页
- [31] 耿德根, 宋建国, 马潮等. AVR高速嵌入式单片机原理与应用. 北京航

- 空航天大学出版社, 2001: 57-76页
- [32] 陶永华, 尹怡欣. 新型PID控制及应用. 机械工业出版社, 1998: 80-85页
- [33] 舒志兵, 刘峻泉等. 闭环伺服系统的数学模型研究. 系统仿真学报. 2002(12): 1611-1613页
- [34] 薛定宇, 陈阳泉著. 系统仿真技术与应用. 北京: 清华大学出版社, 2002: 36-79页
- [35] 姚俊, 马松辉. Simulink建模与仿真. 西安: 西安电子科技大学出版社, 2002: 63-124页
- [36] 张智星著. MATLAB程序设计与应用. 北京: 清华大学出版社, 2002: 94-186页
- [37] [美]约翰·F·加德纳, 周进雄, 张陵译. 机构动态仿真. 西安: 西安交通大学出版社, 2002: 189-265页
- [38] 张静, 张凯, 实时操作系统uC/OS II在ARM7上移植的研究和实现. 计算机工程与应用, 2004(4): 100-102页, 153页
- [39] 杨麟祥, 岳继光, 苏永清. 实时内核uC/OS II在TMS320F240上的移植. 计算机工程与应用, 2004, (8): 10-12页
- [40] Labross J J. 嵌入式实时操作系统uC/OS II(第二版). 邵贝贝等译. 北京: 北京航空航天大学, 2002
- [41] 张立勋, 路敦民, 王岚, 沈锦华. 基于差动机构的五连杆式人机合作机器人的动力学分析. 机器人. 2004(2): 123-126页
- [42] 张立勋, 董玉红, 路敦民. 五连杆式人机合作机器人运动学及动力学研究. 哈尔滨工程大学学报. 2004(3): 337-340页
- [43] 张启先. 空间机构的分析与综合. 机械工业出版社, 1984.3: 12-43页
- [44] 董玉红, 杨清梅. 机械控制工程基础. 哈尔滨: 哈尔滨工业大学出版社, 2003: 17-178页
- [45] 徐滨昌等. 机器人控制工程. 西北工业大学出版社, 1991: 172-186页
- [46] 邓星钟. 机电传动控制. 华中科技大学出版社, 2001: 56-67页
- [47] 李刚. 数字信号微处理器的原理及其开发应用. 天津大学出版社, 2000: 73-97页

- [48] 赵晶. 电路设计与制版Prote199高级应用. 北京: 人民邮电出版社, 2002:1-365页
- [49] 郑雪明. VISUAL C++基础类库参考大全. 北京: 学苑出版社, 1994
- [50] 周鸣杨. VISUAL C++界面编程技术. 北京: 北京希望电子出版社, 2003:48-187页

攻读硕士学位期间发表的论文和取得的科研成果

- [1] 张立勋, 孙卓君, 董九志, 罗红魏. 基于 ATmega128 和 FPGA 的六自由度工业机器人的直流伺服控制器设计. 电子器件(已录用)

致 谢

本论文是在导师张立勋老师的认真指导下完成的。在完成本论文期间，导师不仅传授了学生科学知识，更重视的是开拓了学生的科学研究思路。同时，张老师还不放松对学生进行人生价值的指导教育，这将鞭策学生的一生。导师渊博的知识、敏锐的洞察力、一丝不苟、实事求是工作作风和严谨的治学态度使我受益匪浅。

感谢王岚老师、孟浩和杨勇老师在我读研期间对我的帮助。王老师还特别注重学生动手能力的培养，本文中所涉及到的实验模型和实验过程均是在王老师的指导和帮助下完成的，使学生的动手能力得到了培养。

感谢哈尔滨工程大学机电一体化实验室的所有老师们，我将用我的继续努力来回报你们。

在此，也要向远在家乡的父母致以最深的谢意。论文的完成与他们多年来给我在生活上无微不至的照顾，精神上义无反顾地支持是分不开的。

感谢在本文形成过程中给予了帮助的所有人，我将继续努力。争取人生的更高峰。