

工业机器人控制系统的开放性与实时性研究

韦文求, 谢存禧, 张铁, 庄焕伟

(华南理工大学机械工程学院, 广州 510640)

摘要:针对工业机器人系统对开放性和实时性能的需求,从硬件和软件两方面详细讨论机器人控制器在开放性和实时性上的要求以及目前的研究进展,并从这两方面构建了基于 ARM7微处理器运动控制板的 SCARA 机器人的控制系统,充分利用嵌入式技术、CAN 总线技术、SoC 技术等多项技术,同时将 RTAI(Real Time Application Interface)和 uClinux 相结合组成双核嵌入式实时系统,保证系统具有较强的开放性和硬实时性,最后通过实验验证了系统的可行性和较好的实时性能。

关键词:开放式控制系统;分层结构设计;硬实时性;RTAI

中图分类号: TP391.7 **文献标识码:** A **文章编号:** 1671- 3133(2007)03- 0109- 04

Study on open and real-time performance of industry robot control system

Wei Wen-qiu, Xie Cun-xi, Zhang Tie, Zhuang Huan-wei

(College of Mech. Engineering South China Univ. of Tech. Guangzhou 510640, CHN)

Abstract: The request of the open architecture robot controller and its research development recently from the viewpoint of the software and hardware is discussed based on the needs for the open and real-time performance of industry robot system. And an open architecture scara robot control system is presented which includes hardware and software based on S3C44B0X microprocessor. Some new technologies are utilized in the system, such as embedded technology, CAN technology, SoC (System on a Chip) technology and so on. RTAI (Real Time Application Interface) and uClinux are transplanted and integrated to this robot control system as an embedded and real-time operation system. Those assure this system is open and hard real-time. Finally, the feasibility and real-time performance of this system is confirmed by the experiment.

Key words: The open control system; Multi-layer architecture design; Hard real-time performance; RTAI

0 引言

随着计算机集成制造系统 (CMS) 和柔性制造系统 (FMS) 的发展,越来越迫切地要求工业机器人系统能与现代化工厂的其他自动化系统成为一体,同时随着生产技术的不断更新,以及对机器人系统研究的需要等,这一切都要求机器人系统具有较强的开放性。在实际的应用中,人们总希望机器人的控制系统和结构在某种程度上,为了适应市场的变化而调节生产线和技术需求,可以按照自身和总体系统的功能要求来进行扩展和修改;同时,因为工业生产的自身特殊性——较高实时性、突出的安全性等特性,因此这些特殊要求必须在机器人的控制结构的软件体系中得到实现。由此可见,机器人的控制系统在硬件和软件结构上都必须具有开放性,同时还要保持有较高的实

时响应。

目前,开放式控制系统的研究已成为自动控制领域的一个热门研究方向,并且在国内外取得了进展和成功。美国和欧共体相继推出了 NGC (The Next Generation Work-station/Machine Control Architecture Program) 和 OSACA (Open System Architecture for Control Automation system) 计划,以提高他们的控制系统具有更高的开放性和控制水平,而日本企业也联合推出了 OSEC (Open System Environment for Controller) 计划,并企图使其成为该领域的标准^[1]。中国科学院自动化研究所和上海交通大学等科研院校在该领域也都展开了积极的研究,如沈阳自动化研究所开发了基于 PC 的开放式机器人,充分利用了 PC 机的可扩展性和开放性,具有完备的软件开发环境和丰富的软件资源,以及良好的通讯功能,用户基础广泛。随着嵌入式

处理器的性能和速度的不断提高,嵌入式技术和 SoC (System on a Chip)的不断进步,为开放式控制器的进一步发展提供了更大的空间和可行性。

本文介绍以 ARM7 为运动控制核心,以 uClinux 作为嵌入式操作系统,同时将 RTAI(Real Time Application Interface)和 uClinux 相结合组成双核系统,既满足了开放式的需求,同时又保证系统的硬实时性。

1 系统结构

依据模块化和分层结构的思想,本文构建了以 ARM7 为核心的运动控制板的 SCARA 机器人的开放式控制器。整个系统采用模块化和分层结构设计,根据需要可以配置和扩展不同轴数的机器人系统,该控制器硬件主要是基于斯迈迪公司的 SMP860 运动控制板^[3],并移植了 uClinux 操作系统,同时可通过 CAN 总线与上位 PC 机通讯,由于采用了嵌入式和 SoC 技术,保证了系统的可靠性,缩小了控制系统的体积。

该开放式控制系统的结构如图 1 所示。主要基于 George Saridis 教授提出的分层递阶控制结构以及模块化的思想而设计的,参照组织层、协调层和执行层的智能控制机器结构^[4],该控制系统分为系统层、运动控制层、伺服控制与数据采集层,这样有利于系统整体的开放性和模块化设计,解决了底层的效率和上层与整个工业生产线的协调控制之间的矛盾。

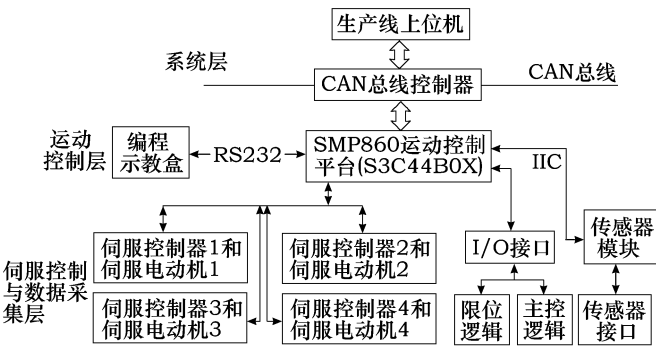


图 1 基于 ARM 的开放式机器人控制系统结构

系统层负责整个系统的在线监控,以及与整个生产线的其他机器人进行数据交换和协调运作,同时还为离线编程以及后续开发提供开发环境。系统层与运动控制层之间通过标准的工业 CAN 总线进行通讯。CAN (Controller Area Network) 总线是一种有效支持分布式控制或实时控制的串行通信网络。采用带 SPI 接口的独立 CAN 控制器 MCP2515 来给 S3C44BOX 扩展 CAN 接口,同时采用 MCP2551 做为 CAN 总线收发器,以保证运动控制层与上位机通讯的稳定性和效率,尽可能降低硬件电路的复杂性。

运动控制层负责机器人整体的运动规划,机械手等运动分析以及各轴之间的插补计算等控制算法的计算。运动控制层主要由 SMP860 运动控制板组成,并将 uClinux 操作系统移植到板上的 S3C44BOX 中,以组成嵌入式独立系统结构。同时通过 RS232 串口与编程示教盒进行通讯,编程示教盒主要完成在线的示教、编程及信息的输入和显示。

伺服控制与数据采集层负责电动机的伺服控制以及传感器与 I/O 的信息采集与处理,其主要包括伺服控制器、伺服电动机、传感器处理模块、I/O 接口模块。伺服控制器采用 YASKWA 的 Σ -II 系列 SGDM-02A,该系列控制器有三种工作模式:速度控制模式、位置控制模式、力矩控制模式。本系统使用位置控制模式,以提高系统整体的位置精度,整个系统采用传统的 PID 控制。电动机采用安川的 SGMPH-02A1A21,该系列的电动机外形尺寸扁平,需要的安装空间较小,满足机器人的要求。虽然驱动器带位置控制环,但为了提高系统的位置精度,将编码器信号通过驱动器反馈到运动控制板的编码器输入端,以便更好地、实时地取得各关节的位置信息,同时也可在软件体系中动态地更改位置和速度增益 (K_p 、 K_v) 值。该开放式控制器的 I/O 信息主要包括各轴的正、反限位信号、伺服报警信号、原点输入信号、复位信号以及各重要的控制开关信号等。传感器模块主要负责采集外部传感器的信号,并通过 IIC (I^2C) 总线传到 S3C44BOX 的 IIC 接口。

2 软件体系结构

机器人控制器的软件体系结构建立在 uClinux 和 RTAI 双内核组成的实时多任务操作系统上,整个系统依然遵循分层和模块化的设计思想,主要包括系统层和应用控制层。该软件体系结构的搭建主要包括:uClinux 系统的裁减和移植、RTAI 内核在 uClinux 系统上的移植、软件搭建、运动控制模块的构建。

2.1 实时系统和应用环境的构建

机器人的控制器是一个多任务的、高精度要求的系统,要求控制器的软件系统具有多任务、实时响应的特点^[5]。目前在这方面的研究主要是集中在基于 PC 上的实时系统的构建,如 RTlinux、为 Windows NT 添加实时模块 RTX 等^[6]。而以上的几种实时系统由于系统太大以及不支持在不同 CPU 上移植等特点很难应用到嵌入式系统上。近年来对嵌入式 Linux 进行实时性改进是嵌入式操作系统领域的一个研究热点,而实时应用接口 RTAI 正是在这种情况下产生的。

RTAI采用双内核方法,不直接使用 Linux的任何功能,而是把需要高度时间精度的工作写成一个驱动程序的形式,然后直接用时序芯片所产生的中断调用这个驱动程序。RTAI在 Linux上定义了一组实时硬件抽象层(RTHAL),其将 RTAI需要在 Linux中修改的部分定义成一组程序界面,RTAI只使用这组界面和 Linux沟通^[7]。uClinux是一个源码开放的操作系统,面向没有 MMU(Memory Management Unit)的硬件平台,去除了普通 Linux内核中的虚拟内存管理部分^[8],并且 uClinux提供了完整的 TCP/IP协议栈,为嵌入式系统提供了强大的网络支持。其主要是面向嵌入式运用,并保留了 Linux大多数的优点,且结构高度优化、代码紧凑,但是其本身没有更多的关注实时性,难以满足机器人系统的实时响应和精度要求。

从前面的分析可以看出,RTAI是基于普通 Linux内核,相对于嵌入式应用其内核过于庞大;而 uClinux本身并没有更多地关注实时问题。因此,将 RTAI和 uClinux相结合,采用双内核的设计方案,既满足了嵌入式应用的需求,又保证了机器人系统的硬实时性以及开放性。

基于 uClinux 和 RTAI的双内核实时系统整体结构模型如图 2 所示,主要包含了基本内核 uClinux、实时内核 RTAI、硬件抽象层 RTHAL、基于 S3C44B0X的 SMP-860 运动控制板。其中基本内核(uClinux)和实时内核(RTAI)分别处理非实时和实时任务的调度和执行,而实时任务和非实时任务之间信息的交换要通过管道(FIFO)或共享内存(MBUFF)来实现。当实时任务运行时,基本内核被硬件抽象层屏蔽。即实时内核将基本内核作为优先级最低的一个任务来运行,只有在没有实时任务运行的时候才予以调度。实时内核 RTAI是动态加载的,只有在实时任务激活时,基本内核才被实时内核接管,否则基本内核通过 RTHAL透明访问硬件,就像 RTHAL不存在一样。且通过调度处理基本内核中断和实时中断,当中断到来时,实时内核判断它是基本内核中断还是属于实时中断,分别进行处理。同时,实时中断可以抢先基本内核任务的执行。

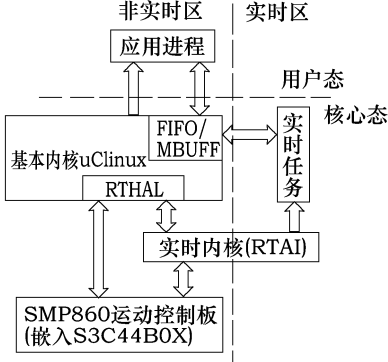


图 2 双内核实时系统嵌入式实时 uClinux架构

双内核的移植工作主要包括 bootloader的编写与移植、uClinux和 RTAI内核的编译与移植。其中 bootloader亦称为系统引导程序,类似于 PC的 BIOS 主要任务是固化 uClinux和 RTAI内核与文件系统、系统初始化、设定内核加载方式以及地址映射与分配等任务。主要的过程如下:如果上电检测启动参数正确,则调用 decompress_kernel()以解压 uClinux内核,并调用 call_kernel()初始化与加载内核,否则显示加载错误。本文采用的是 RTAI-2.4.10,编译环境为 Red Hat 9.0下的 arm-linux交叉编译环境,编译 uClinux内核并为之打上 RTAI内核包。

2.2 基础运动控制系统

该控制器的基础运动控制系统软件体系结构主要包括系统管理和通讯模块、初始化与数据处理模块、运动控制与轨迹插补模块、动态跟踪与检测模块四个部分,如图 3所示。

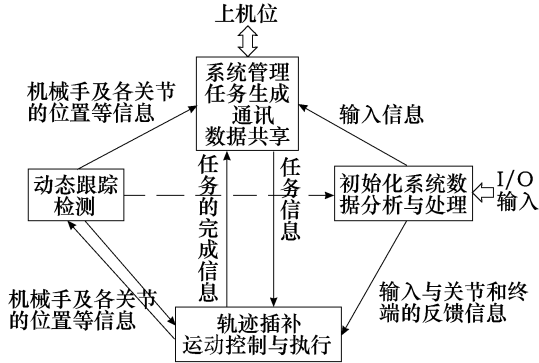


图 3 运动控制系统软件体系结构图

系统管理和通讯模块主要负责和上位机以及和模块之间的通讯,同时管理整个系统的信息调度和共享,以及示教编程,通过 SPI接口的 MCP2515 独立 CAN控制器与上位机进行通讯;初始化与数据处理模块主要初始化运动控制板和伺服控制器,并对输入的数据和参数进行处理后传送到系统管理和通讯模块;运动控制与轨迹插补模块负责根据要求完成机械手的轨迹插补算法,包括直线、圆弧、S曲线插补,并实时反馈位置和速度到动态跟踪和检测模块,以及实时根据数据处理模块的数据更正轨迹和插补路径;动态跟踪与检测模块主要动态实时跟踪各轴的转角和位置数据缓冲区,并反馈到系统管理和数据处理模块,取得机械手的末端当前位置坐标,以进行实时的调整插补轨迹。

3 系统实时性分析

机器人控制系统对实时性要求非常严格,主要表

现为:1)实时周期性的系统状态检测、读取关节编码器的位置信号,为了达到较高精度,这些任务被重新调度的周期要求很短,因此系统对周期性的实时任务的支持很重要;2)轨迹规划与运动插补等控制进程要求尽可能快的被调用,以执行正确的控制策略和精度。鉴于以上分析需要对系统的时间分辨率和系统响应时间进行测试。

3.1 系统的时间分辨率测试

分别在 uClinux和 RTAI/uClinux内核下利用定时函数设计周期性的高优先级任务,每个周期切换一次运动控制板的 I/O 口电平,找出各自可分辨的、稳定的方波最小周期^[9],则系统可分辨的最小任务周期即为方波周期的一半。在处理器中等负荷条件下,在 RTAI/uClinux内核下测得的不同周期下的 I/O 输出波形,分析得出在定时周期大于等于 55μs时输出波形较稳定,且可分辨;同样的方法得出在 uClinux内核

下,可分辨的最小任务周期为 25ms 最后的试验结果如表 1所示,而 55μs的周期分辨率足以满足控制器任务调度的周期(一般的都在 1ms左右^[6])。

表 1 系统的时间分辨率测试结果

内核	方波最小周期 T	时间分辨率 t=T/2
uClinux	50.10ms	25ms
RTAI/uClinux	110.05μs	55μs

3.2 系统响应时间的测试

在不同的系统内核,以及处理器中等负荷的情况下,运行一个不做任何处理的简单用户程序,系统内部的定时器分别记录运行这个简单用户程序前和执行该应用程序 main()时的时刻,计算二者的时间差就可以得到系统响应时间。表 2是系统响应时间的测试结果,其中每项数据都是每 100次对简单应用程序的系统响应时间。

表 2 系统响应时间的测试结果

μs

试验方式		实验次数										
		1	2	3	4	5	6	7	8	9	10	均值时间
(100次系统响应时间 t)	uClinux	13101	12510	14102	13560	13489	13520	12891	14103	13481	13458	13421.5
	RTAI/uClinux	2510	2768	2674	2458	2684	2768	2568	2510	2410	2561	2599

从上面的测试数据看,100次 uClinux系统响应的平均时间为 13421μs RTAI/uClinux系统的 100次系统响应的平均时间为 2599μs 因此 uClinux与 RTAI/uClinux一次系统响应时间大致分别为 134μs和 26μs 性能较好,满足控制器的精度和实时性需要。

4 结论

本文在以 S3C44B0X为处理器的 SMP860运动控制板上,构建了实时的 SCARA型软硬件开放的机器人控制器系统的整体结构,并采用了 RTAI/uClinux双内核组成实时嵌入式系统,通过实时性试验分析表明,系统具有较高的实时性,控制系统在软硬件体系上都采用了分层和模块化设计,用户可以方便地进行扩展和修改,整个系统具有良好的实时性和可扩展性。

Symposium on Implicit and nonlinear Systems Ft Worth, Tx, 1992

[5] Stephen Balakirsky, Chris Scrapper, Elena Messina Mobility Open Architecture Simulation and Tools Environment[J]. KIMAS 2005, April 18 — 21, 2005: Waltham, MA, USA, 1—2

[6] 张广立,付莹,杨汝清,等.基于 Windows NT 的开放式机器人实时控制系统[J].上海交通大学学报, 2003, 37(5): 724-728

[7] 崔珂,陈伟,等.一种改进嵌入 Linux实时性的新型方案[J].电子工程专辑, 2005(1—2).

[8] 李善平,刘文峰,等. Linux与嵌入式系统[M].北京:清华大学出版社, 2003

[9] 范海涛,王树民.基于 RTAI的 uClinux硬实时性能的实现[J].电力自动化设备, 2006, 26(3): 67—68

作者简介:韦文求,硕士研究生,研究方向:机器人及其自动化。
谢存禧,博士生导师,研究方向:机器人设计理论及工程应用。
张铁,副教授,研究方向:机器人设计理论及工程应用。
庄焕伟,硕士研究生,研究方向:机器人设计理论及工程应用。

参考文献:

[1] 王天然,曲道奎.工业机器人控制器的开放式体系结构[J].机器人, 2002, 24(5): 256—260

[2] 孙斌,杨汝清.开放式机器人控制器综述[J].机器人, 2001, 23(4): 374—378

[3] 深圳市斯迈迪科技发展有限公司. SMP860用户手册 v1.2, 2005

[4] Saridis G. N. Architecture for Intelligent Controls IEEE