

DOI:10.3969/j.issn.1001-4551.2016.12.010

基于 Machinekit 的多轴控制器设计*

谢 锴, 颜钢锋*

(浙江大学 电气工程学院, 浙江 杭州 310027)

摘要:针对传统多轴控制器的结构、扩展性和实时性存在的问题,对实时 Linux 操作系统、Machinekit 控制软件以及 EtherCAT 协议等方面进行了研究,对 Machinekit 控制软件构成进行了介绍,对实时 Linux 操作系统的选取原因进行了阐述,对多轴控制器的结构、实时性和稳定性进行了分析,在保证多轴控制器实时性和稳定性的前提下,提出了一种基于 Xenomai 和 Machinekit 的多轴控制器软件架构,并将实时以太网 EtherCAT 协议作为传输协议应用到控制器中,利用系统对伺服轴的控制进行了功能验证以及利用 Wireshark 捕捉控制报文来对系统的实时性进行了分析测试。研究表明,基于 Machinekit 和 EtherCAT 的多轴控制器能够完成对多轴的基本控制工作。

关键词: Machinekit; Xenomai; EtherCAT; 多轴控制器; 软件设计

中图分类号: TP29; TH39

文献标志码: A

文章编号: 1001-4551(2016)12-1471-06

Design of multi-axis controller based on Machinekit

XIE Kai, YAN Gang-feng

(College of Electrical Engineer, Zhejiang University, Hangzhou 310027, China)

Abstract: Aiming at the structure, expansibility and real-time problems of traditional multi axis controller, the real time Linux system, Machinekit control software and EtherCAT protocol were studied. The composition of Machinekit control software was introduced, and the reasons for the selection of real time Linux operating system were described. At the same time, the structure, real time and stability of multi axis controller were analyzed. On the premise of ensuring the real-time property and stability of the multi axis controller, a software architecture of multi axis controller based on Xenomai and Machinekit was proposed. And the real time Ethernet protocol (EtherCAT) was applied to the controller. To verify controller's function, three servo axes were used, and Wireshark was used to capture the control message to analyze the real-time performance of the system. The results indicate that the controller based on Machinekit and EtherCAT is able to complete the basic control of multi-axis.

Key words: Machinekit; Xenomai; EtherCAT; multi-axis controller; software design

0 引 言

近些年来,特别是在进入 21 世纪以来,工业机器人大量应用在汽车制造,电子生产等现代制造业中。工业机器人的控制系统作为机器人的核心部分,显得非常重要,传统的控制系统主要是“PC 机 + 控制卡”

或者是“DSP + FPGA”的方案,系统性能受到很大的限制,同时,轴的控制数量不会超过 6 个,在需要控制的轴数目增多时,控制系统的成本会大大的增加,对扩展性提出了很大的要求^[1]。实时以太网在近些年发展非常迅速,由于实时以太网的数据包容量高,传输距离远,开放性好等特点^[2],被越来越广泛的应用在多轴控制器领域。目前国际上有很多种实时以太网, Ether-

收稿日期: 2016-06-12

基金项目: 国家自然科学基金资助项目(61175106)

作者简介: 谢 锴(1991-),男,河南商丘人,硕士研究生,主要从事工业以太网和机器人技术方面的研究. E-mail: xk2010@163.com

通信联系人: 颜钢锋,男,教授,博士生导师. E-mail: ygf@zju.edu.cn

CAT(ethernet for control automation technology) 技术是由德国 Beckhoff 公司提出的一种解决方案,此解决方案基于标准的以太网技术,使用特殊的以太网数据帧类型(0x88A4),系统配置简单,数据传输全双工,高速,高效,总线利用率较高,数据有效率达到 90% 以上^[3]。

Machinekit 是一个开源控制软件,用来运动控制,典型应用在机器人,3D 打印机,数控加工系统等方面。Machinekit 最初是来自 LinuxCNC,和 LinuxCNC 相比, LinuxCNC 主要应用在数控加工等领域^[4], Machinekit 应用更为广阔,不仅仅对于本地机器的控制,也可以使用 Machinekit SDK 对远程机器的控制, Machinekit 极大地丰富了 LinuxCNC 的应用^[5]。

本研究以 PC 为平台,在 Debian 操作系统上,移植实时框架,来保证控制系统的实时性和稳定性。同时,笔者在实时操作系统上,利用 Machinekit 作为控制软件,同时使用 IgH 的 Etherlab 作为主站模块,使用 EtherCAT 作为通讯协议,实现具有 EtherCAT 通讯功能的多轴控制器。

1 Machinekit 简介

Machinekit 在 2014 年从 LinuxCNC 分离开来,基本架构特性和 LinuxCNC 基本上是一样的,框架如图 1 所示。

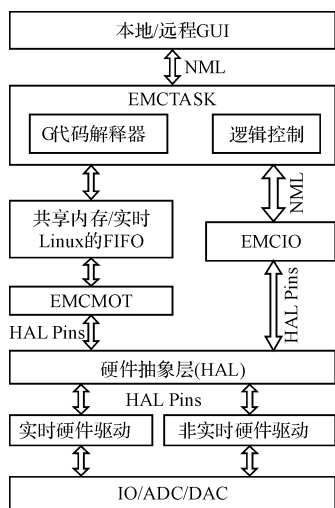


图 1 Machinekit 框架

该框架主要有 4 个组成部分:运动控制(EMC-MOT),离散 IO 控制器(EMC IO),任务执行器(EMC-TASK)和用户图形接口(GUI)^[6]。在这 4 个部分中,只有运动控制模块是实时模块,非实时模块之间通过中性消息语言(NML)进行通讯,而实时模块(EMC-MOT)与非实时模块(EMCTASK)之间的通讯是通过

共享内存或者实时 Linux 系统的 FIFO 机制^[7]。

1.1 运动控制器(EMCMOT)

EMCMOT 是唯一的实时模块,实现对路径的规划,正逆姿态算法的计算。这个过程包括,各个轴的位置取样,运动插补,运动规划等。从图 1 中看出,EMC-MOT 通过共享内存或者 Linux 的 FIFO 机制与 EMC-TASK 进行包括运动命令,状态和错误信息在内的信息交互。另一方面通过硬件抽象层与实际硬件驱动进行信息交互。EMCMOT 在整个系统中是被周期性执行的,在每个控制周期或者取样周期内,EMCMOT 对轴位置和速度进行取样,计算误差,根据设定的 PID 计算出下一个控制后期的输出,在每个采样周期进行状态更新,通过共享内存的方式传给 EMCTASK,然后在 GUI 上进行显示。

1.2 离散 IO 控制器(EMC IO)

离散 IO 控制器控制所有的 IO 事件,但是由于 IO 控制器设计高度依赖机器,不同设备的 IO 设计不同,所以 EMC IO 不直接控制实际的物理轴,而是通过硬件抽象层(HAL)进行控制,在 HAL 文件中,描述了应用的 IO 接口,信号等。

1.3 任务执行器(EMCTASK)

EMCTASK 在整个软件框架中,位于 EMC MOT 和 EMC IO 的最上层,使用 NIST RST 库和 NML 类进行编写,主要是进行 G 代码和 M 代码的解释^[8]。任务执行器一方面监控 EMC IO 和 EMC MOT 的状态,通过 NML 发送到 GUI 显示,另一方面接收并分析来自 GUI 的命令,或者是远程 GUI 的命令,并将命令向下层转发。

1.4 GUI 和硬件抽象层(HAL)

和 LinuxCNC 一样,Machinekit 提供一些 GUI 供选择,包括 AXIS,mini 和 tklinuxcnc 等。GUI 使用 NML 与 EMCTASK 通讯,一方面为用户展示系统状态,另一方面可以通过 NML 发送控制命令。GUI 可以是本地的,也可以是远程的。与远程 GUI 通讯分两种情况。第一种,主 PC 上运行 GUI,键盘,鼠标和 GUI 窗体信号通过相关软件转送到远程 PC,在主 PC 上运行 X server,建立 X11 连接,在远程 PC 的 Windows 系统上可以使用类似 Xming 软件进行连接。X11 和 VNC 连接可以通过安全套接字(SSL)来加密,在两者之间进行安全传送。第二种,GUI 程序直接在远程 PC 上运行,通过 NML 与主 PC 通讯,但是需要对配置文件进行相应的修改,如增加主 PC 的 IP 地址等。

硬件抽象层(HAL)提供的用户接口(halui)将

HAL 管脚和 NML 命令进行连接,在硬件和软件之间提供映射,从而保证了系统的硬件无关性,具有较好的可移植性。

2 控制系统软件架构

结合 Machinekit 的特点以及 EtherCAT 实时以太网的要求,同时基于控制器软件设计思路的分层化设计和模块化设计,提出了控制系统的软件架构如图 2 所示。

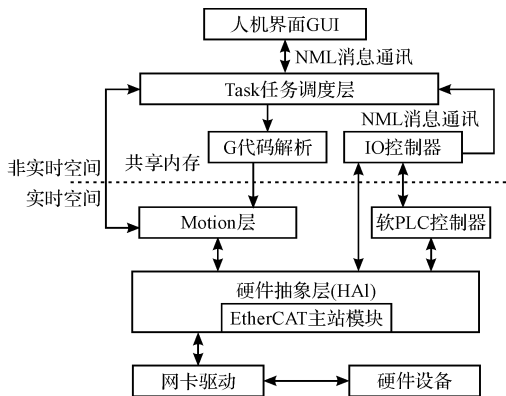


图 2 系统软件框架

系统软件主要分为几个部分:

- (1)非实时空间。主要包括人机界面,非实时传感器 IO 信号的接收处理,实现了多轴的任务调度和路径规划,环境探测,限位开关,急停,指示灯等等;
- (2)实时空间。主要包括 Motion 层,软 PLC 控制层,硬件抽象层。Motion 层主要完成速度规划,运动插补等,软 PLC 控制器部分主要负责实时逻辑控制;
- (3)实时以太网 EtherCAT 主站模块。这里采用 Etherlab 开源主站,其提供了一种在 Linux 上的 EtherCAT Master 解决方案^[9],实现了主站对从站(伺服系统,输入输出模块)的实时控制。

3 系统设计

系统在个人 PC 上来实现,设计内容包括,实时 Linux 系统的实现,EtherCAT 主站的实现,以及将 EtherCAT 主站作为组件嵌入到硬件抽象层。

3.1 实时 Linux 系统的实现

目前实时操作系统有很多解决方案,Windows CE,INtime,RTLinux,RTAI,Xenomai 等,但是对于该系统,希望能够代码开源,这样方便修改代码,使其最大程度上适合硬件,适合的有 RTAI 和 Xenomai,这两种方案均支持 RTDM,其中 RTAI 主要关注点在于提升实时性,Xenomai 主要考虑稳定性,可移植性等^[10]。比

较情况如表 1 所示。

表 1 实时框架 RTAI 与 Xenomai 的特性比较

	RTAI	Xenomai
架构支持	x86, x86-64, PowerPC, ARM, MIPS	X86, x86-64, PowerPC, ARM, PowerPC64, Blackfin
实时空间	Kernel, user	Kernel, user
提供的 API 接口	RTAI, POSIX 1003. 1b, RTDM	Xenomai Native, Posix 1003. 1b, RTAI, VxWorks, pSos +, VRTX uITRON, RTDM

由于 Etherlab 和 Machinekit 对于 Xenomai 的 API 支持以及工业对稳定性的要求,系统采用 Xenomai 实时框架作为解决方案。Xenomai 是一种双内核机制的 Linux 内核扩展,优先级高于 Linux 内核,与 Linux 共享硬件,只有在没有 Xenomai 任务的时候,才会处理 Linux 任务,这种优先级规则保证了实时任务系统的实时性和可靠性^[11]。Xenomai-Linux 的架构如图 3 所示。

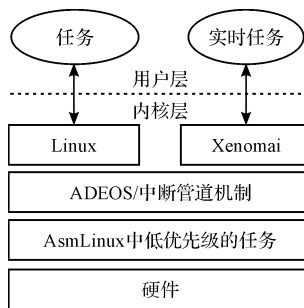


图 3 Xenomai-Linux 软件框架

开发过程主要是对于 Xenomai 在 PC 机上的移植,以及对实时性进行测试。PC 机上装有 Debian 的 Jessie 版本,按照 Xenomai 的说明进行安装,安装的 Xenomai 的版本为 3.8.1-xenomai. x86-amd64。

3.2 EtherCAT 和 Machinekit 的重要配置

(1) EtherCAT 主站。EtherCAT 主站采用开源 Linux 主站代码,在安装配置需要根据自己硬件和系统修改相关参数,同时需要生成基于 Xenomai 的实时库,在 Xenomai 上实现 EtherCAT 主站的软件系统架构如图 4 所示。

从图 4 看出,软件框架主要有两层需要设计,EtherCAT Master 和 Xenomai-Linux,其中 EtherCAT Master 使用 Etherlab 提供的源码,API 层为 Application 提供接口,Master Core 提供协议的解析等。Xenomai-Linux

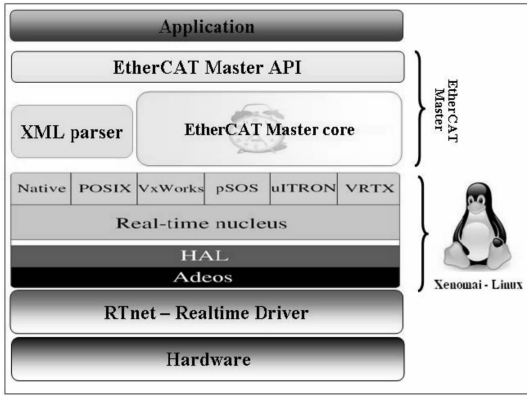


图 4 Xenomai-Etherlab 软件框架

为 Xenomai 部分,通过 Adeos 提供一些实时性保证。

下载解压 Etherlab 源代码之后,配置源代码,生成 Makefile。在配置的时候,需要使能 RTDM,使能周期定时器 hrttimer。然后进行编译内核模块,加载到内核。根据自己的硬件,配置相关的参数。

由于应用层无法直接访问内核层,需要通过构造通道来和主站进行间接通讯。普通用户如果想建立读取通道,必须设置一条设备管理器的规则,来创造设备节点。

比如新建文件/etc/udev/rules.d/99-EtherCAT.rules,在里面写入 KERNEL == " EtherCAT[0-9] * ", MODE = "0664", GROUP = "users"

(2)将 Etherlab 编译为 HAL 组件。Machinekit 是 github 上的一个开源项目,编译安装的时候需要使能 Xenomai,安装的时候,将 source list 中加上 deb.machinekit.io/debian Jessie main 的地址,然后可以直接通过命令行安装。

安装好之后,如图 2 所要求的,需要将 EtherCAT 作为 HAL 组件加入 Machinekit 中,实现细节参考后文。

3.3 运动控制模块与 EtherCAT 的 HAL 配置

运动控制模块是多轴控制器的核心组成,主要完成轨迹规划、正逆运动解算、插补运算、加减速等,本研究针对该系统的三轴实验平台,构建自己的 HAL 文件,示例如图 5 所示。

其中,loadrt 命令为加载实时内核命令。lcec 是由 Etherlab 的主站代码编译出来的组件,需要加载到内核。首先,需要将 ethercat 的配置文档加载进来,loadrt trvkins 为加载正逆运动解算模块,loadrt classicladder_rt 为加载梯形图,然后加载功能模块,将梯形图变量映射到轴变量。

3.4 软 PLC 模块设计

和 LinuxCNC 类似,Machinekit 中也存在软 PLC 模

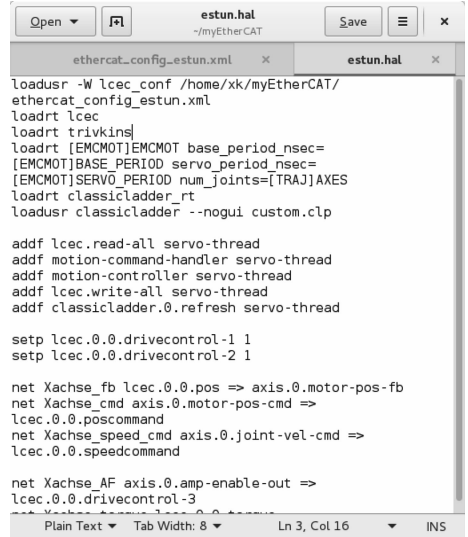


图 5 HAL 配置

块,实现多轴控制中的逻辑控制。系统设计中将 PLC 变量分为两个层次,一个为高级指令,包括多轴的紧急指令,出错应急指令,异常捕获指令,执行需要较高的权限。另一个为普通指令,如普通的电磁阀,开关等指令。

正如图 5 中要求,设计需要在 HAL 文件中将梯形图加载,然后将信号和管脚进行映射。

梯形图示例如图 6 所示。

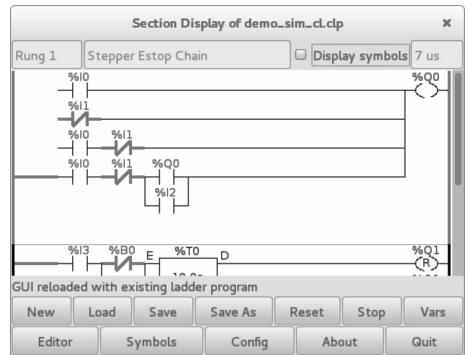


图 6 梯形图示例部分

3.5 HAL 下的 EtherCAT 组件开发

正如前面的 HAL 示例文件所示,需要将 EtherCAT 作为组件加入到 HAL 映射中,HAL 本质就是使用 Xenomai 的实时函数,构造具有管理功能的软件框架。而文件中也能看出,需要将输入引脚、输出引脚、函数、线程进行连接,然后进行加载,为实现 HAL 下的 EtherCAT 主站,需要将 Etherlab 提供的主站代码,编译为 HAL 下的实时内核模块,在 HAL 文件中进行加载,就可以实现 EtherCAT 通讯。

在 HAL 文件中需要加载配置文件,配置文件主要是根据伺服驱动器的设备描述文件来进行编写的,比如实验平台使用 Estun 的伺服,以及一些 Beckhoff 公

司的 IO 设备,其中 Estun 的配置文档示例如图 7 所示。

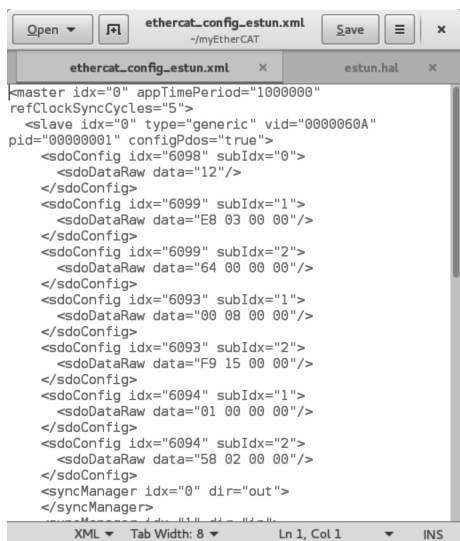


图 7 设备描述文件配置

4 实验验证

4.1 系统组成

为测试系统的兼容性,本研究搭建了实验平台,硬件连接图如图 8 所示。

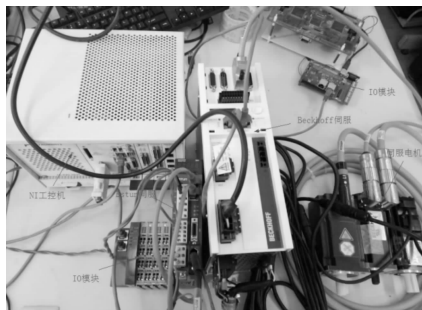


图 8 硬件连接图

平台分为 3 个部分:①非实时部分、开关、传感器等。②主 PC,采用装有 Jessie 的 PC 机。③实时单元执行器部分,包括 Beckhoff 伺服驱动器和电机,Estun 伺服驱动器和电机,IO 模块。

4.2 实验结果

(1)Xenomai 测试。在安装好 Xenomai 之后,对其进行测试。

首先,在/usr/bin/下的 realtime start 来启动实时内核,进行系统时钟测试 (clocktest),结果如图 9 所示,从图 9 中可以看出,CPU 分为 4 核,时钟漂移最坏为 32 ns。

然后进行 latency 测试,在/usr/xenomai/bin/下的 latency,如图 10 所示。

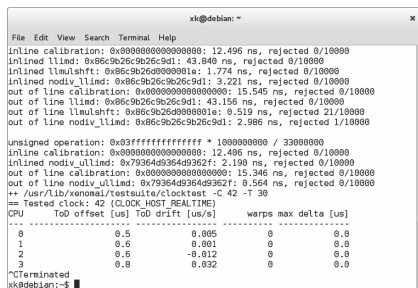


图 9 Xenomai 的 clocktest

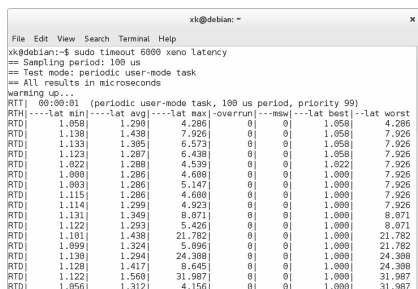


图 10 用户模式下的 Latency 测试

从图 10 中看出,在用户模式下,平均延时在 1 ms。在使用 Machinekit 之前,使用 latency-test 来进行 HAL 延时测试,如图 11 所示。

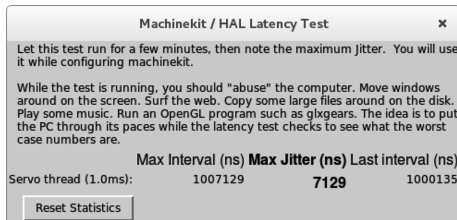


图 11 Machinekit Latency 测试

测试的时候,需要进行拖动窗体,打开浏览器,复制大文件等操作,来保证测试的合理性和最坏情况,从而可以在 Machinekit 配置的时候使用这些数据。

从图 11 可以看出,最大的抖动为 7 129 ns,在微秒级别。

(2)控制器功能测试。界面在原有的基础上进行修改,为三轴控制器,Machinekit 实验界面如图 12 所示。

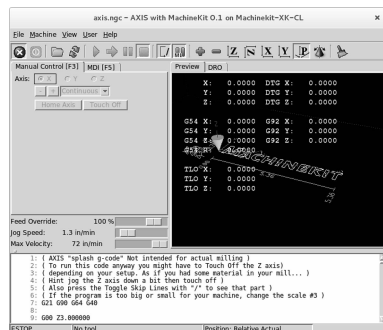


图 12 Machinekit 实验界面

下面就一个例子对功能实现进行阐述,比如使终端按照 X 轴移动,需要以下几个步骤。首先,图 12 中 GUI 给出按照 X 轴移动信号,信号通过 NML 发送到 EMCTASK 进行 G 代码的解释。其次,解释出的命令通过共享内存发送到 EMCOT 实时模块。最后,将命令通过 HALPins 发送到 EtherCAT 模块。具体来说,由于伺服采用的是 CoE(CanOpen over EtherCAT)协议,通过 EMCOT 对配置文件(图 7)中的控制字 0x6040 进行写十进制 15,来使能电机,并且设定为速度模式,对控制字 0x60FF 写 Jog 速度,即可实现对电机的速度控制。

实验验证,控制器可以控制多轴,实现 Machinekit 对多轴的基本控制。

(3)多轴之间同步性测试。本研究就搭建的实验平台,在 Debian 上安装 wireshark,进行抓包分析。在 Xenomai-Linux 的主站控制下,数据负载较小时候,截取部分数据,分析如表 2 所示。

表 2 低负荷情况下同步时间统计

报文离开网卡时间/ms	报文回到网卡时间/ms	时间差/ms
0	0.065	0.065
1.981	2.051	0.07
3.969	4.049	0.08
5.862	5.928	0.066
7.818	7.881	0.063
9.822	9.903	0.081
11.718	11.781	0.063
13.725	13.793	0.068
15.647	15.723	0.076
17.56	17.63	0.07

报文 2 ms 循环一次,延时在 70 us 左右,延时局限在网卡的能力和处理器处理能力上,现在实时性保证在 μs 级别,这样的实时性能和通讯速度已经非常出色。

5 结束语

本研究将开源控制技术 Machinekit 和工业实时以太网 EtherCAT 技术相结合,提出了一种基于 Xenomai-Linux 的 EtherCAT 多轴控制器方案。考虑到控制器的实时性,在 Linux 上移植 Xenomai 实时框架,保证了控制器所采用的系统实时性,其次,由于软件设计的分层化和模块化,将 EtherCAT 主站编译成模块嵌入到 Machinekit 组件中,考虑到伺服设备的多样性,对设备描述采用 XML 文件,利用 EtherCAT 的兼容性保证了

系统的扩展性。该方案实时性强,扩展性好,对不同种类设备兼容性高,由于系统软件的开源性,方便再开发,满足现代工业控制对多轴控制器的要求。实验结果表明,基于 Machinekit 的多轴控制器能够完成控制工作。

参考文献(References):

- [1] 李春木,章明众,郑新武,等. EtherCAT 总线在伺服运动控制系统中的应用[J]. 机电工程. 2011,28(11):1336-1338.
- [2] 李具备,栾勇,王超,等. 基于 AM3358 处理器的嵌入式实时 EtherCAT 主站的构建[J]. 组合机床与自动化加工技术. 2015(5):1-5.
- [3] 刘艳强,黄帅,马秋霞. 基于工业以太网 EtherCAT 的 DCS 控制系统设计[J]. 制造业自动化. 2010,32(11):21-23.
- [4] 樊是星. 基于 LinuxCNC 控制器的六自由度并联机构人机交互系统设计[D]. 镇江:江苏科技大学机械工程学院,2014.
- [5] WINGS E, MÜLLER M, ROCHLER M. Integration of real-time Ethernet in LinuxCNC[J]. **The International Journal of Advanced Manufacturing Technology**. 2015, 78 (9-12):1837-1846.
- [6] SHACKLEFORD W P, PROCTOR F M, MICHALOSKI J L. The neutral message language: A model and method for message passing in heterogeneous environments[C]. **Proceeding of the 2000 World Automation Conference**, New York: National Institute of Standards and Technology,2000.
- [7] STAROVEŠKI T, BREZAK D, UDILJAK T. LinuxCNC - the enhanced machine controller: application and an overview[J]. **Technical Gazette**,2013,20(6):1103-1110.
- [8] MARIETTA M. Draft volume I of next generation workstation/machine controller (NGC) specification for an open system architecture standard[Z]. 1992.
- [9] 高美原,秦现生,白晶,等. 基于 ROS 和 LinuxCNC 的工业机器人控制系统开发[J]. 机械制造,2015,53(10):21-24.
- [10] MOON Y, KO N Y, LEE K, et al. Real-time EtherCAT Master Implementation on Xenomai for a Robot System[J]. **International Journal of Fuzzy Logic & Intelligent Systems**. 2009,9(3):244-248.
- [11] KOH J H, CHOI B W. Real-time performance of real-time mechanisms for rtai and xenomai in various running conditions[J]. **International Journal of Control and Automation**. 2013,6(1):235-246.

[编辑:张豪]

本文引用格式:

谢 锴,颜钢锋. 基于 Machinekit 的多轴控制器设计[J]. 机电工程,2016,33(12):1471-1476.

XIE Kai, YAN Gang-feng. Design of multi-axis controller based on Machinekit[J]. Journal of Mechanical & Electrical Engineering, 2016,33(12):1471-1476.

《机电工程》杂志: <http://www.meem.com.cn>