

基于 RTX 的步进电机寿命实验

李寅龙 张新玉 张 凤

(北京空间机电研究所 北京 100094)

摘要: 步进电机在现代控制中发挥着越来越重要的作用,为了保证产品的可靠性,步进电机的寿命的评估也变得尤为重要。基于 Windows 的评估系统能够很好的解决这个问题,但是由于 Windows 操作系统的实时性较差,而步进电机的控制需要精确的时序,所以由其产生的驱动信号时序无法满足要求。采用了 Ardence 公司的基于 Windows 的硬实时解决方案 RTX,并结合某项目步进电机寿命实验,总结出了 Windows 环境下 RTX 实时应用系统的设计方法与开发流程。实验证明基于 RTX 产生的驱动信号满足步进电机的运动要求,并且能够不受外部事件的干扰,重复性良好,说明该方法是实时的、稳定的、可靠的。

关键词: 步进电机;RTX;Windows 环境;寿命实验

中图分类号: TP273 **文献标识码:** A **国家标准学科分类代码:** 510.8040

Life test of stepping motor control base on RTX

Li Yinlong Zhang Xinyu Zhang Feng

(Beijing Institute of Space Mechanics & Electricity, Beijing 100094, China)

Abstract: Stepping motor is playing an increasingly important role in modern control. In order to guarantee the reliability of the product, the stepping motor's life becomes especially important. The evaluation system based on Windows can solve the problem very well. However due to the Windows operating system has poor real-time performance, the driving signal timing produced by it can't meet the requirements of stepping motor which requires precise timing. This paper adopted the hard real-time solution RTX based on Windows which is produced by Ardence. Combining with the experiments of project about stepping motor life, the paper summarized the design method and development process of RTX real-time application system. It is proved by experiments that drive signal generated based on RTX can satisfy the demands of stepping motor. Additionally, it is not affected by the external events and has good repeatability. As a result, the method is real-time, stable and reliable.

Keywords: stepping motor; RTX; Windows; lift test

1 引言

步进电机是一种将电脉冲信号转换成角位移的执行元件,当系统将一个电脉冲信号加到步进电机定子绕组上,转子就转过一个固定角度(步距角);当一组电脉冲按某一相序加到步进电机时,转子沿某一方向转动的步数就等于电脉冲个数。因此,改变输入电脉冲的数目就能控制步进电机转子角位移的大小;改变电脉冲的通电相序,就能控制转子角位移的方向^[1]。

随着步进电机驱动技术和数字控制技术的发展,步进电机在雷达制导、激光探测、红外扫描、火控系统军事领域广泛应用^[2-3]。步进电机的特点是体积小,寿命长,寿命通常取决于轴承寿命。由于步进电机在现代控制中发挥着

越来越重要的作用,为了保证产品的可靠性,步进电机的寿命的评估也变得尤为重要,所以需要步进电机进行寿命实验。进行步进电机寿命实验的方案有多种,本文结合了一个步进电机寿命实验的项目,为了方便调整实验参数,存储实验结果,绘制实验波形,采取了在 Windows 平台下控制步进电机进行寿命实验的方案。

Windows 操作系统为桌面操作系统,并没有对实时性进行专门的优化,所以 Windows 定时器的精确性不高,并且当用户使用鼠标或键盘时,Windows 定时器的响应会受到影响。如果基于 Windows 定时器产生控制时序的话,会导致步进电机运行不稳,所以基于 Windows 定时无法满足步进电机对驱动信号时序的要求,而 Ardence 公司的 RTX 系统可以满足该定时要求^[4-6]。

2 RTX 系统介绍

Windows 系统利用普通 PC 为硬件平台,软件开发手段比较丰富,通用性较强,但实时性较差^[7]。而步进电机的控制,需要精确的时序控制,并且输出的时序必须是连续完整的,不能被其他 Windows 程序(比如操作键盘等)所打断,所以只使用 Windows 系统无法满足步进电机的控制要求。

本文选择应用嵌入式开发软件,修改 Windows 内核,在 Windows 内部构建实时嵌入式系统,搭建实时仿真开发环境的方法。该方法既利用了 Windows 强大的图形界面功能,又通过构建实时嵌入式系统弥补了 Windows 实时性不足的缺点。

本文选择基于 Ardence 公司的 RTX 开发基于 Windows 嵌入式实时环境^[8-10]。RTX 首先对 Windows 内核体系进行了延拓,修改并扩展了整个 HAL(硬件抽象层),从而使其以并列子系统的形式实现独立的内核驱动模式。由于 RTSS 的线程优先级高于 Win32 应用线程和系统线程调度的优先级,故 RTSS 线程的各项实时性能可不受 Win32 背景任务的影响,真正避开 Windows 内核,实现实时处理。其体系结构如图 1 所示。其具有的特点为:

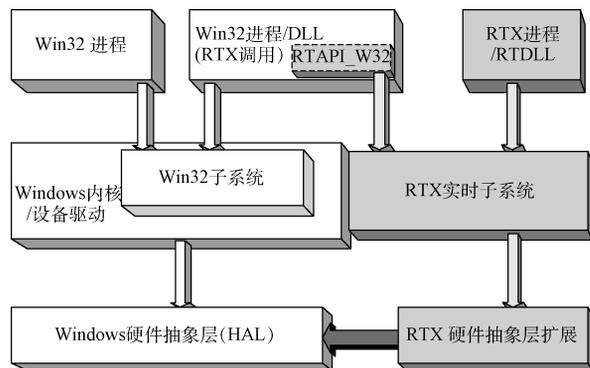


图 1 RTX 体系结构

1) 使用 Windows 系统的调度策略,对自己的实时线程进行严格调度管理;同时,RTX 线程(包括中断服务、延迟调用)总是优先于 Windows 线程取得调度权。

2) 通过扩展的 HAL,RTX 有着自己中断管理机制,而且能够直接访问 I/O 硬件端口。

3) 对所有线程执行抢占的实时调度,支持着 128 个优先级,严格控制各个优先级线程间的切换。

4) 提供与 Win32 环境兼容的实时编程接口,并且支持 Win32 的 API,实现了一套实时函数的扩展,包括中断管理函数和直接硬件访问函数。

计算机识别一个外部事件到做出响应的时间是非常重要的指标,图 2 给出了 RTX 实时系统和 Windows 操作系统响应时间测试结果,可以看出,RTX 实时系统的响应时

间无论是最大值还是平均值都比 Windows 操作系统的响应要小得多,而且 RTX 的响应时间比较稳定^[11]。

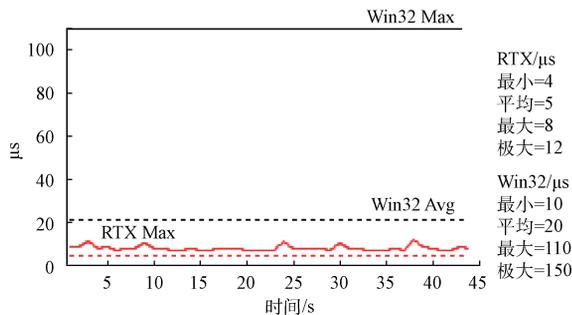


图 2 响应时间测试结果

3 系统方案

为了操作方便,便于实验的参数调节和观看步进电机的运动情况,监测实验状态,存储实验数据,步进电机寿命实验采用了 PC 内的 PCI 板卡作为步进电机的控制器,Windows+RTX 控制软件通过对 PCI 板卡^[12-13]的操作,实现对步进控制控制时序的输出,并采集霍尔位置传感器的信号,在测试模式下,系统还具有失步检测功能,具体的方案如图 3 所示。

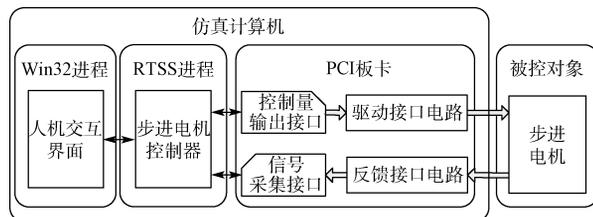


图 3 步进电机控制总体方案

该系统主要由实时仿真计算机、PCI 板卡和被控对象组成。实时仿真计算机是步进电机寿命实验系统的主体,它由人机交互软件接口、步进电机控制器和 PCI 驱动程序组成。人机交互软件接口用于设计人员设置电机运动参数,控制电机运动状态和分析运动结果等功能;步进电机控制器用于接收上位机控制参数,定时输出步进电机控制时序,接收电机反馈信号;PCI 驱动程序用于与 PCI 板卡进行通信^[14-15],从而输出控制时序并接收反馈信号,此功能集成在 RTSS 进程中。PCI 板卡里面集成了驱动接口电路和反馈接口电路;驱动接口电路用于为步进电机控制信号提供驱动电流;反馈接口电路用于步进电机的反馈通路。

步进电机一步起速,由于此时负载的角速度为 0,根据运动方程式可知,需要的力矩可由式(1)得到:

$$T = \frac{J\pi\theta \cdot f^2}{180} \times 2 \quad (1)$$

式中: $\theta=0.9^\circ$ 为电机步距角, f 为转速, J 为转动惯量, T 为电机输出力矩。

从式(1)中可以看出随着步进电机启动转速的增加,需

要的电机输出力矩呈平方方式增加,如果当负载转换到电机轴上的转动惯量 J 比较大时,随着启动转速的增加,需要的输出力矩增加的非常快。步进电机的输出的力矩是有限的,所以步进电机的启动转速需要进行限定,否则如果当电机启动频率过快时,可能会由于电机驱动力矩不够,而导致电机失步。

根据上述分析,需要对电机进行启动加速控制,同理,也需要进行停止减速控制。所以为了能够让步进电机运行稳定,并且步数准确,需要对驱动时序进行规划,添加加速和减速过程^[16],具体的转速分布如图 4 所示。

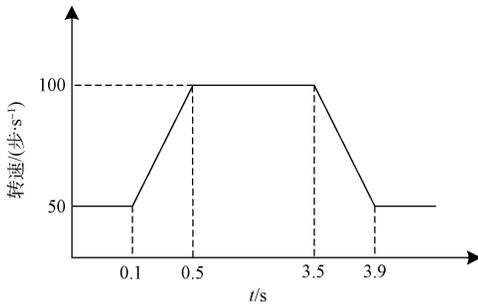


图 4 步进电机转速分布

从图 4 中可以看出,步进电机一步起速,从 0 达到 50 步/s,然后转速从 50 步/s 均匀加速达到 100 步/s,当需要停止时,转速从 100 步/s 均匀减速达到 50 步/s 后最终停止。

步进电机需要满足一定的驱动时序才能工作,电机的运动速度是通过对 A、B、C、D 的驱动时序的输出频率进行控制的,具体的驱动时序如表 1 所示。

表 1 步进电机驱动时序

驱动端 时刻	A	B	C	D
T0	1	0	0	0
T1	0	1	0	0
T2	0	0	1	0
T3	0	0	0	1

控制步进电机的加速、减速或平稳速度是通过控制输出端 A、B、C、D 的驱动时序的频率来实现的,当增大四相输出端的驱动时序的频率时,步进电机则加速运行;当减小四相输出端的驱动时序的频率时,步进电机则减速运行。在本文中,步进电机的驱动时序是通过基于 RTX 编写的实时模块输出的。

4 软件实现

4.1 软件总流程

根据 RTX 和 Windows 的特性,将软件划分成了两部

分:基于 Windows 的非实时进程和基于 RTX 的实时进程。两大进程之间通过高速 IPC 通信和同步机制,实现数据交换。实时进程由 RTX 实现,主要完成霍尔器件信号接收、步进电机信号的发送;非实时进程由 Windows 实现,主要完成电机数据的处理、图形显示及参数输入功能。

非实时部分包括参数设置界面、主流程、电机数据显示界面和与 RTX 进程通信的线程。其中参数设置界面主要用于进行步进电机运动参数设置及电机运动状态设置;电机数据显示界面实现当前步进电机运动步数,运动方向,运动圈数等运动信息,每 100 ms 刷新一次;非实时部分程序创建了与 RTX 进程通信的线程、共享内存和事件,向 RTX 进程发送电机运动参数、电机运动开关指令,并接收 RTX 进程反馈的电机运动信息。

实时部分包括主程序、PCI 驱动程序、指令处理程序和定时器中断服务程序。主进程完成定时器创建及定时中断服务程序挂接、打开非实时部分创建的共享内存和事件等;PCI 驱动程序完成了与 PCI 板卡的通信功能,可以实现控制量的输出和反馈量的输入;指令处理程序接收来自非实时部分的指令,解析指令并执行;定时器中断服务程序根据指令处理程序设定的电机运动参数,输出不同频率的步进电机的驱动时序,从而达到控制步进电机的转速的效果。程序流程如图 5 所示。

采用 Visual C++6.0 和 RTX SDK 为开发工具,首先建立一个 MFC AppWizard 工程来作为 Win32 进程环境,然后建立一个 RTX AppWizard 工程来作为 RTSS 进程环境,在 Win32 工程下利用 RtCreateProcess() 进行调用 RTSS 进程编译生成的 .rtss 文件,此时完成两个进程的建立,进程间采用共享内存机制进行通信。

4.2 软件具体实现

本节给出了软件部分模块的具体实现方法,分别为实时部分的主函数模块、PCI 驱动程序模块、中断服务模块和非实时部分的通信模块和使用界面。

4.2.1 实时部分模块

1) 主函数模块

RTX 的主函数中使用了 RtOpenEvent 和 RtOpenSharedMemory 函数与 MFC 界面程序进行了连接,实现了事件和存储空间的共享。为了能够与 MFC 界面程序进行了指令消息传递,本文中自定义了一些消息,比如,MSG_GETMOTORPAR,MSG_SETMOTORPAR 等,并且使用循环结构等待 MFC 界面程序发送消息;当接收到消息指令后,通过解析共享内存空间的数据来获得 MFC 界面程序对步进电机运动参数的配置。

为了能够实现平稳的加速或减速曲线,本文使用 RtCreateTimer 函数创建了定时周期为 100 μ s 的定时器,并与中断服务程序挂接。从而产生了步进电机的驱动信号。

2) PCI 驱动程序模块

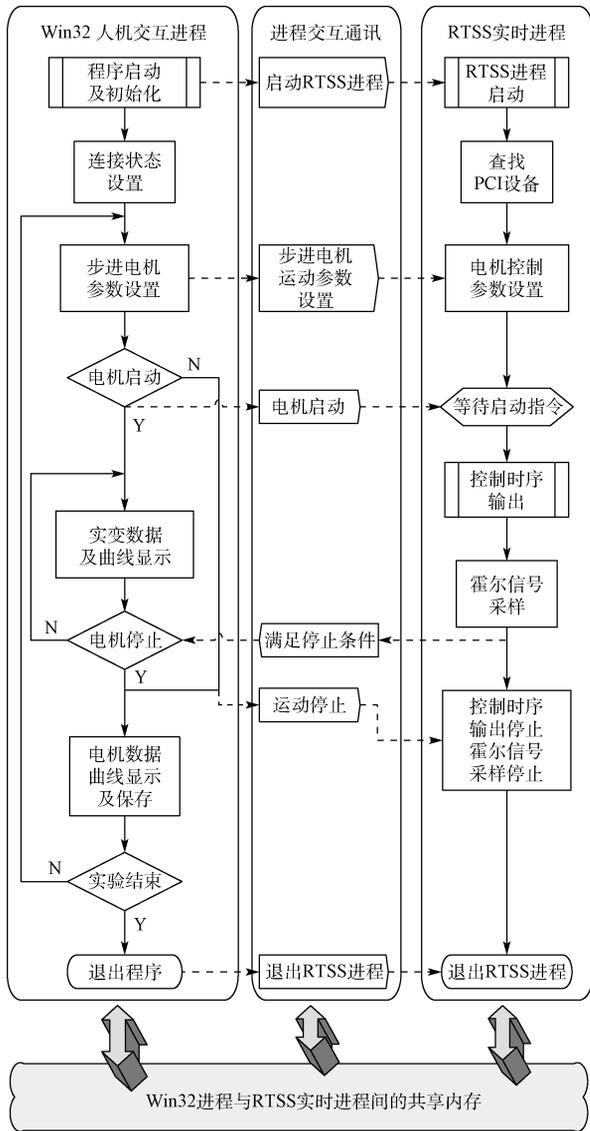


图 5 步进电机寿命实验程序流程

RTX 支持对硬件的 I/O 和内存直接操作,所以可以通过 RTX 自带的驱动程序对 PCI 板卡进行驱动。PCI 板卡一般采用 I/O 地址映射和内存地址映射两种方式,RTX 提供了一系列输入输出函数直接操作 I/O 端口与内存。

首先通过 RtGetBusDataByOffset 函数扫描板卡所在总线上的信息,从而得到本板卡的端口基址与范围。在得到板卡的基址之后,使用 RtWritePortXXX 和 RtReadPortXXX 函数对板卡上的相应寄存器进行读写操作。由于这些操作时在 RTX 下对硬件的直接访问,不会受到 Windows 系统的影响,所以实时性能得到了很大的提高。

3) 中断服务模块

在中断服务模块中,通过判断在主函数中设置的电机运行参数,通过 PCI 驱动程序模块对步进电机运动进行

控制。

图 6 为终端服务程序的流程图,从图 6 中可以看出程序实现了步进电机的加速运行,减速运行和平稳运行、正向运行、反向运行,通过读取霍尔信号来判断步进电机是否转完一圈,并且判断步进电机是否失步。根据 MFC 界面程序设定好的步进电机运行参数,比如,运行状态、运行圈数和运行方向等,来对步进电机进行控制。

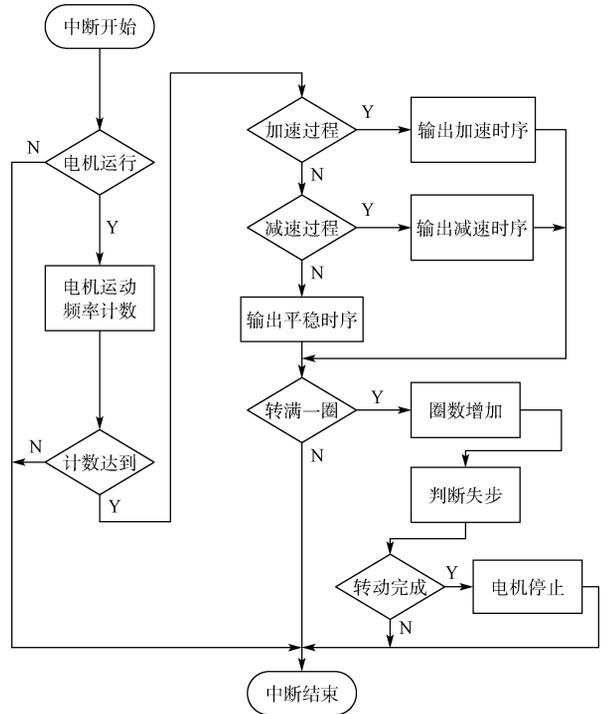


图 6 中断服务程序流程

4.2.2 非实时部分模块

非实时部分通信模块的功能是,接收界面输入模块的电机运动参数,把电机运动参数通过共享内存的方式传递给实时部分主函数模块,并通过消息发送控制指令;接收实时部分主函数模块发来的电机运动数据,发送给非实时部分的电机数据显示界面模块。

实现的主要步骤是通过 RtCreateSharedMemory 函数和 RtCreateEvent 函数创建用于与 RTX 部分通信的消息和共享内存,当有电机参数需要更新到 RTX 模块时,把电机参数填写到共享内存相应的位置,然后用 RtPulseEvent 向 RTX 部分发送控制指令 MSG_SETMOTORPAR;当更新界面的定时触发后,用 RtPulseEvent 向 RTX 部分发送 MSG_GETMOTORPAR,然后等待数据返回消息,当返回消息来到后,则读取共享内存里的电机运动数据,并调用电机数据显示界面模块。

步进电机寿命实验软件的使用界面如图 7 所示。可以分别对某个电机进行运动设置,并且可以满足 4 个步进电机同时进行寿命实验。



图 7 步进电机寿命试验软件的使用界面

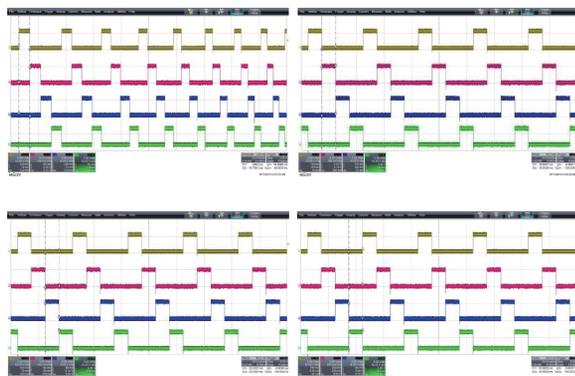


图 8 步进电机控制时序输出

5 实验验证

为了验证本文中的步进电机驱动技术的准确性和可靠性,基于该技术开发了步进电机驱动器,并对某项目的步进电机进行了寿命实验。图 8 为输出的步进电机控制时序图。

从图 8 中可以看出步进电机的控制时序是通过 A, B, C, D 4 项交替输出信号来实现的。通过使用 RTX 定时输出信号,所以输出信号的脉冲宽度为 10 ms,输出信号准确,满足步进电机运行要求;并且能够长时间工作,满足步

进电机寿命实验的要求。

对基于 RTX 定时和基于 Windows 定时两种情况下的输出波形进行了重复性实验,如表 2 所示,可以得出基于 RTX 定时的时序输出的误差在 0.04% 以内,即误差在 4 μs 以内,而基于 Windows 定时的时序输出的误差在 2% 以内,即误差为 200 μs 以内。

表 2 不同定时类型输出波形的脉冲宽度

次数类型	1	2	3	4	5	6	7	8	9	10
RTX	9.999	9.997	9.998	10.003	9.999	10.002	10.001	10.001	9.998	9.999
Windows	10.154	9.839	9.968	10.187	9.898	9.970	10.045	10.185	9.834	9.957

考虑到外部的因素,比如操作人员移动鼠标或者键盘输入等事件,由于移动鼠标或者键盘输入会产生输入中断,使得 Windows 系统需要对其进行处理,当使用 Windows 进行电机驱动,外部事件与电机驱动信号输出相当于串行处理,外部事件会打断驱动信号输出,导致驱动信号输出有秒级的偏差,无法驱动步进电机运行;当使用基于 RTX 定时的方式,RTX 系统单独占用 CPU,不会被外部事件打断,所以外部事件不会对驱动时序的输出产生影响,保证了存在干扰的情况下的实时性。

对基于 RTX 定时的启动加速控制进行了测试,如图 9 所示,从图 9 中可以看出,启动时,输出波形的频率为 50 Hz,并随着时间增加而增加,最终达到稳定频率 100 Hz;对于减速过程也是同样的道理。

进电机控制要求。并使用该步进电机寿命实验软件进行了长达 10 d 的步进电机寿命实验,结果表明该软件满足步进电机对驱动时序的要求,能够精度并且稳定地完成寿命实验。

6 结 论

本文使用了 RTX 的实时环境,实现了步进电机控制,并给出了软件部分模块的实现流程。结合某项目的步进电机寿命实验,给出了系统的实验结果,说明了使用 RTX 进行步进电机控制的方法是可行的,并且由于基于 RTX 定时输出时序的精度高,可精确到 4 μs,所以该系统也适用于其它需要在 Windows 平台下精确定时的引用。

参考文献

- [1] 汤拓. 小型步进电机控制系统设计[J]. 信息通信, 2015(3):61-61.
- [2] 王帅,朱娟. 小型位移台的步进电机控制系统设计[J]. 信息化研究, 2012(2):26-29.
- [3] 高琴,刘淑聪,彭宏伟. 步进电机控制系统的设计及应用[J]. 制造业自动化, 2012, 34(1):150-152.
- [4] 黄键,宋晓,薛顺虎. RTX 平台下实时仿真系统的设计方法[J]. 计算机应用与软件, 2009, 26(4): 167-169.



图 9 步进电机启动加速控制时序输出

上述实验表明,使用 RTX 能够满足步进电机的精确控制,并且不受外部事件的干扰,能够满足寿命实验的步

- [5] 许志,唐硕.基于 RTX 实时环境的快速原型系统设计[J].西北工业大学学报,2010,28(2):176-181.
- [6] 杨维,石德乾,赵凯,等.基于 Windows XP+RTX 的火控调试平台软件设计[J].火炮发射与控制学报,2013(4):30-34.
- [7] 王亮,邓红德,康凤举,等.基于 RTX 的航电综合模拟系统应用研究[J].仪器仪表学报,2011,32(11):2475-2480.
- [8] 高泽东,李建军,高教波.基于 RTX 的实时伺服控制系统[J].计算机应用,2011,31(S2):212-215.
- [9] 张前福,王华闯,许俊平,等.基于 RTX 的实时处理系统设计[J].Microelectronics & Computer,2012,29(8):58-61.
- [10] 王伟志,裘著有,王贲.基于 RTX 实时仿真系统的设计与实现[J].电子设计工程,2013,21(12):24-27.
- [11] 夏红伟,凌明祥,曾庆双,等.基于 RTX 的卫星姿轨控系统地面实时仿真系统[J].计算机仿真,2006,23(9):40-42.
- [12] 张彪,宋红军,刘霖,等.基于 PCIe 接口的高速数据传输系统设计[J].电子测量技术,2015,38(10):113-117.
- [13] 王爽,李全令,相朋举.基于“柔性测试”技术的 CPCI 模块测试平台 CPCI[J].国外电子测量技术,2015,34(4):10-13.
- [14] 傅瑶,薛旭成,郝伟,等.基于 Visual C++ 的 1780 定时器采集卡内部晶振定时的实现[J].国外电子测量技术,2014,33(11):82-85.
- [15] 付宁,徐东东,杨易,等.基于 PCI9054 的 PCI 主模式板卡的设计[J].电子测量技术,2016,39(1):93-98.
- [16] 孙孔政,徐抒岩,曹小涛,等.基于矩频曲线的步进电机加减速控制[J].微电机,2014,47(1):81-85.

作者简介

李寅龙,1986 年出生,硕士,北京空间机电研究所工程师,主要研究方向为伺服控制与应用。

E-mail:liylnong2006@126.com

(上接第 123 页)

- [9] 王立莹,王俊芳,曹琨,等.VLAN 及在千兆以太网 MAC 中的实现[J].无线电工程,2013(1):1-3.
- [10] 王祖云,杨思国,王建伟,等.嵌入式 LwIP 协议栈的移植与测试研究[J].计算机与数字工程,2014(2):272-275.
- [11] 张松,李筠.FPGA 的模块化设计方法[J].电子测量与仪器学报,2014,28(5):560-561.

作者简介

郭桦炜,1990 年出生,厦门大学硕士研究生,主要研究方向为核磁共振仪器研制。

E-mail:ghwwz@163.com

郑振耀(通讯作者),1966 年出生,副教授,主要研究方向为计算机工业过程控制、电子技术应用、核磁仪器研制等。

E-mail:zzy@xmu.edu.cn