文章编号:1671-4598(2019)05-0122-05 DOI:10.16526/j.cnki.11—4762/tp.2019.05.028 中图分类号:TH166 文献标识码:A

基于 RTX64 的软运动控制器设计与实现

杨 啸[1],李 迪[1],李 松[2],王世勇[1]
(1. 华南理工大学 机械与汽车工程学院,广东 广州 510640;
2. 深圳市旗众智能科技有限公司 运动控制研发中心,广东 深圳 518100)

摘要:针对运动控制器实时性、开放性、柔性化、可配置的需求,实现了基于 RTX64 的软运动控制器;根据多任务对实时性需求不同,采用“Windows7 + RTX64”的方案,将实时任务运行在 RTX 实时子系统内;采用分层架构模式、功能模块化的设计思想,使用共享内存、环形缓冲区分别作为进程、线程间的通讯方式,同时利用互斥体、临界区域保证数据交互的正确性;通过实时工业以太网 EtherCAT 总线技术实现软控制器与从站设备的通讯;讨论了软控制器的事件流与数据流模型;最后,设计了以软控制器为主控单元的测试平台;实验表明,基于 RTX64 的软运动控制器达到了通用运动控制平台的高性能要求,具有较高的稳定性、开放性、扩展性。

关键词:软运动控制器;RTX;分层架构;功能模块化;EtherCAT

Research and Implementation of Soft Motion Controller Based on RTX64

Yang Xiao[1], Li Di[1], Li Song[2], Wang Shiyong[1]
(1. School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China;
2. Motion Control Research Center, Shenzhen City Public Intelligent Technology Co., Ltd., Shenzhen 518100, China)

Abstract: The soft motion controller based on RTX64 is implemented to meet the requirements of real-time, openness, flexibility and configurability. According to the different real-time requirements of multi-task, the scheme of “Windows7 + RTX64” is adopted and the real-time task is run in the real-time subsystem. Applying the idea of layered architecture and function modularization, the shared memory is used in inter-process communication with mutex, the ring buffer is used in inter-thread communication with critical section and the communication between controller and slave devices is realized by EtherCAT bus technology. Furthermore, the task and data flow are also studied in detail. Finally, the platform based on controller is designed. Experimental results show the controller fulfills the high performance requirements of the general motion control platform, and achieves high stability, openness and expansibility.

Keywords: soft motion controller; real time eXtention; layered architecture; function modularization; EtherCAT

0 引言

“中国制造 2025”对制造业提出了“数字化、网络化、智能化”的新要求[1],运动控制器作为制造业的核心技术之一,已从封闭式逐步发展为开放式[2]。开放式运动控制器主要有三种架构:PC 嵌入 NC、NC 嵌入 PC、软控制器[3]。在智能制造的新时期,软控制器能充分利用 PC 机的性能优势,以模块化软件实现实时控制功能,具有可重构、可扩展、柔性化等特点,发展前景良好,成为当前的研究热点之一,Brecher 等[4]以典型厂商、研究项目为例详细地介绍了开放式控制系统的发展与现状;朱达宇等[5]在 Linux 平台下采用 RTLinux 实时扩展,构建全软件数控系统;王普等[6]研究了基于 RTX 的全软件数控系统,分析测试了系统的实时性能,并构建原型系统;Yu 等[7]分析了传统数控系统的体系架构与现场总线的特点,采用组件技术设计了基于现场总线的开放式数控系统;宋利利等[8]以三轴仿真转台为实际应用背景,针对实时性、稳定性的系统需求,设计了基于 RTX 的实时控制软件;毕鲁雁等[9]设计了基于 RTX 的工业机器人控制系统,为机器人控制算法与功能扩展提供了基础平台;Wu 等[10]通过“PC+Windows 实时扩展”的方式探究了开放式软数控系统的实施方案,并验证了系统的实时性与功能。

以上研究主要集中在 32 位平台下的专用控制,对非标设备控制研究较少。然而随着控制需求的不断提高,定制化产品、个性化设备需求的不断出现,研究基于高性能 64 位平台的通用软控制器具有一定的现实意义。

本文提出以“Windows7 + RTX64”为主控单元,结合 EtherCAT(Ethernet for Control Automation Technology)总线协议,设计并实现了 64 位平台下的通用开放式软运动控制平台,并提供软件开发工具包(Software Development Kit, SDK),可适用于不同的解决方案,为促进控制器实时性、扩展性、柔性化可配置的发展提供参考。

收稿日期:2018-11-15; **修回日期**:2018-12-10。
基金项目:广东省科技计划资助项目(2017B090913002,2017B090914002);智能制造综合标准化与新模式应用项目。
作者简介:杨 啸(1994-),男,河南驻马店人,硕士研究生,主要从事高性能嵌入式控制系统方向的研究。
李 迪(1965-),女,山东青岛人,教授,博士生导师,主要从事嵌入式系统、自动控制和机器视觉方向的研究。

1　软运动控制器系统设计

如图 1 所示，软运动控制器是由实时任务与非实时任务组成的多任务混合系统，采用“Windows 7 ＋ RTX64 3.0”设计方案，在 Windows 下运行非实时任务，在 RTX 实时子系统（Real－time Subsystem，RTSS）中运行实时任务。采用 EtherCAT 总线协议实现工控机与驱动器从站、I/O 从站等硬件设备的通讯。

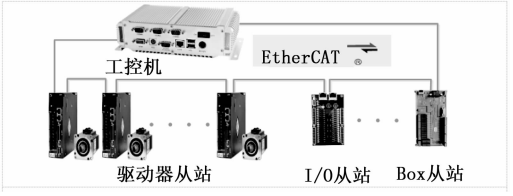

图 1　软运动控制器系统方案

作为通用运动控制平台，软运动控制器的模块化功能主要包括：参数配置模块、运动控制模块、安全控制模块、从站控制模块、反馈功能模块。在不同的工艺需求中，可基于软运动控制器设计个性化的解决方案，实现运动逻辑自定义，具有可配置、可重构、可扩展的优点，如图 2 所示。

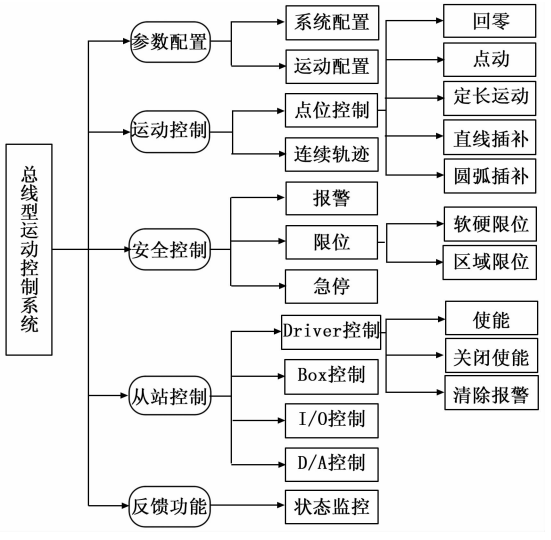

图 2　软运动控制器功能模块图

2　软运动控制器软件实现

2.1　RTX64 系统简介

RTX64 是 Interval Zero 公司开发的运行在 64 位平台下的实时子系统。RTX64 通过扩展 Windows 硬件抽象层（Hardware Abstraction Layer，HAL）实现从 Windows 隔离系统资源，独占处理器，保证 RTSS 和 Windows 之间的中断隔离，确保了实时指令、RTSS 调度任务等只能在 RTSS 独占的处理器上运行[11]。

RTSS 提供了基于优先级的抢占式调度，线程优先级可设置为 0－127，共 128 个等级，其中 127 为最高优先级。RTX 线程的优先级始终高于 Windows 线程[9]。同时，RTX 拥有自己的时钟和定时器服务，时钟精度为 100 ns，确保实时任务的可靠性。RTSS 可通过 IPC 对象实现与 Windows 系统间的数据交互，通过内核同步机制保证数据同步与通讯。

RTX64 作为 Windows 系统实时扩展的首选方案，可充分利用 Microsoft Visual Studio 强大的开发、调试功能，降低开发难度、提高开发效率。

2.2　分层架构设计

根据运动控制器混合多任务的特性，采用模块化设计思想，将事件触发、对实时性要求较低的人机交互界面（Human Machine Interface，HMI）运行在 Windows 应用层，将周期性执行、对实时性要求较高的运动控制内核、EtherCAT 主站等运行 RTSS 中。

如图 3 所示，软运动控制器分层架构从上至下依次为应用层、数据交互层、实时控制层、设备层。

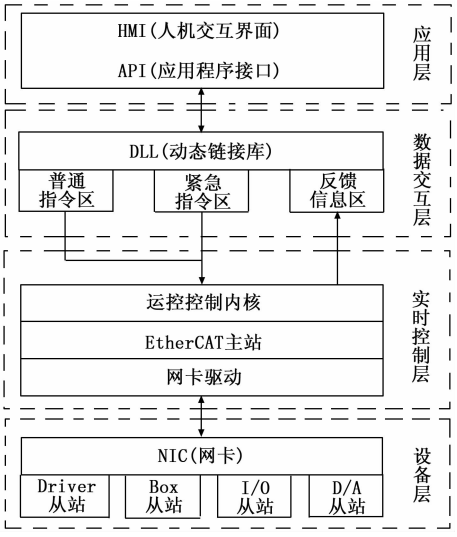

图 3　软运动控制器分层架构方案

1）应用层包含 HMI 与应用程序接口（Application Programming Interface，API）。通过 HMI 可将 API 指令信息传递至数据交互层。

2）数据交互层包含动态链接库（Dynamic Link Library，DLL）与三块共享内存空间：普通指令区、紧急指令区、反馈信息区。根据执行指令的紧急与否设计了快慢双缓冲的环形队列方案：普通指令区存放非紧急指令，如参数设置、直线插补；紧急指令区存放紧急指令，如急停、在线改变目标位置；紧急指令区优先级高于普通指令区。此外通过反馈信息区，可监控运动控制器的状态，如实际位置、实际速度、IO 状态等。

3）实时控制层包含运动内核、EtherCAT 主站、网卡驱动。运动内核的主要功能为轨迹规划；EtherCAT 主站负责运动内核与 EtherCAT 从站间的数据运输，如：驱动器从站的过程数据对象（Process Data Object，PDO）。具体

流程为：运动内核将应用层的运动指令通过解析、预处理、插补计算等处理后，交由 EtherCAT 主站。主站将运动内核输出数据打包，通过网卡发送下行数据帧至各个从站模块。同时，主站也将从站上行数据帧反馈至运动内核。

4）设备层由网卡与从站模块组成，在设计的运动控制器中采用了总线型拓扑结构，从站模块包括：Driver 从站、Box 从站、I/O 从站、D/A 从站等。

2.3 进程间通讯

分层架构表明软运动控制器是由人机交互界面、运动内核、EtherCAT 主站三个进程组成。其中，人机交互界面运行在 windows 系统中，运动内核、EtherCAT 主站运行在 RTSS 中。三者协同工作，通过使用共享内存实现进程间通讯，以互斥体确保进程间交互数据的正确性，避免竞争条件的产生。

1）人机交互界面与运动内核间的通讯：

HMI 根据指令时效性高低发送至不同的指令共享内存。在运动内核解析任务中，紧急指令优先于基本指令被解析。同时运动内核周期性地将实时运动数据更新至反馈共享内存，HMI 采用循环扫描的方式获取反馈信息，并对变动信息进行更新显示。

2）运动内核与 EtherCAT 主站间的通讯：

以数据结构体的形式实现交互，在结构体内部设计两块环形缓冲区。运动内核将指令发送至环形缓冲区 rbSend，同时将标记 bMCtoMaster 置为真，EtherCAT 主站收到标记为真的消息后，取出运动内核指令，并将标记还原。同样的，运动内核收到 EtherCAT 主站反馈数据标记后，从环形缓冲区 rbRece 中获取反馈数据，并将反馈标记清零，如图 4 所示。

```
Struct stCommData
{
  ......
  BOOL bMCtoMaster;
  CRingBuffer<BYTE,SEND_BUFSIZE>rbsend;
  BOOL bMastertoMC;
  CRingBuffer<BYTE,RECE_BUFSIZE>rbRece;
  stMailBoxSDOMailBoxSDO;
  ......
}
```

图 4 通讯结构体设计

2.4 线程间通讯

环形缓冲区具有先进先出（First In First Out，FIFO）的特点，在多线程异步协作或生产者消费者模型中被广泛使用[12]。采用环形缓冲机制能有效解决数据生产与消费速度不匹配的问题、增强多线程的并行性能[13]、减少“数据饥饿”、避免“内存碎片”的产生。模板是 C＋＋泛型编程的重要思想，具有灵活性强、代码复用的优势。采用模板类设计环形缓冲区，在构造函数内进行成员变量的初始化，在析构函数内清理成员变量，确保资源的自动管理。

软运动控制器将基于模板类的环形缓冲区与临界资源结合使用，确保多线程间的数据保护与通讯，如图 5 所示，CRingBuffer 类实现了存放不同数据类型的环形缓冲区，可从缓冲区内添加、获取一组或多组数据，并根据实际需求更改缓冲区大小。

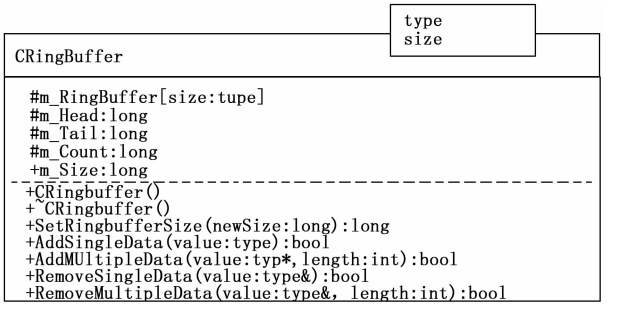

图 5 基于模板类的环形缓冲区 UML 类图

软运动控制器采用了多线程流水线设计模式。由于控制器具有点位运动与连续轨迹运动两种模式，故采用多生产者单消费者模型，在该模型中，当前线程既作为上游线程的消费者，又作为下游线程的生产者。若当前线程判断存在以下情况之一：（1）上游生产者缓冲区为空；（2）下游消费者缓冲区空间不足；（3）插补计算轴的引用计数非 0，则放弃当前线程执行权。否则，当前线程获取上游缓冲区数据，经过解析、计算、封装处理后，打包发送至下游缓冲区。此时本次线程调度完成，由运行状态切换至休眠状态，等待下一次被唤醒。

2.5 运动内核多线程调度方案

软运动控制器在实时内核层采用多线程并行工作方式实现强实时任务的设计。运动内核作为控制器的核心，主要包括以下线程：主函数线程、上位机指令解析线程、连续轨迹预处理线程、插补计算线程、主站通讯线程、状态反馈线程。根据各线程任务的紧急程度与重要性，设置了不同的调度周期与线程优先级。此外，为保证下游数据冗余、避免数据饥饿，将线程在运行状态下设置为执行不同的次数。如表 1 所示为运动内核线程调度方案。

表 1 运动内核线程调度方案设计

线程任务	周期/ms	执行次数	优先级
主站通讯线程	0.5	1	126
插补计算线程	0.5	2	125
连续轨迹预处理线程	1	4	120
上位机指令解析线程	1	从站总数	117
状态反馈线程	1	1	110
主函数线程	1	1	1

主函数线程作为第一个被创建和最后一个退出的线程，负责启动时软控制器的初始化工作与退出时软控制器的清理工作；上位机指令解析线程以动态的从站数量作为每周

期扫描次数，避免 HMI 指令流量过高造成指令阻塞。并将解析后的点位运动发送至插补计算线程、连续轨迹运动发送到预处理线程；连续轨迹预处理线程负责前瞻与小线段平滑过渡，将处理后的指令发送至插补计算线程；插补计算线程采用 S 型算法计算出每个周期内的路径规划位置，并发送至主站通讯线程；主站通讯线程负责运动控制内核与 EtherCAT 主站间的数据交互。

2.6　EtherCAT 通讯

EtherCAT 是德国倍福自动化有限公司推出的实时工业以太网协议，具有实时性强、拓扑结构灵活、有效数据率高的优点[14]。软运动控制器基于 RTX64 在 Windows 平台下开发了 EtherCAT 主站程序，采用 CoE（CANOpen over EtherCAT）协议用于伺服驱动器应用层通讯，通讯周期为 1ms，调度优先级为 127，控制模式为周期性同步位置模式（Cyclic Synchronous Position Mode，CSP）。

EtherCAT 主、从站通过状态机切换可进入运行（OP）状态[15]。对伺服驱动器还需通过控制字切换至使能状态。在运行过程中，运动内核的主站通讯线程通过共享内存实现与主站的通讯，如将接收过程数据对象（Receive Process Data Object，RxPDO）、IO 输出、模拟量输出等指令信息发送给主站，并从主站获取发送过程数据对象（Transport Process Data Object，TxPDO）、IO 输入、手摇脉冲发生器（Manual Pulse Generator，MPG）位置等。

3　软运动控制器流程设计

控制器在运行过程中事件流的执行与数据流的传递过程，如图 6 所示。具体工作流程可描述为：（1）启动在 windows 系统下运行的 HMI 后，HMI 创建其与运动内核交互的共享内存，并启动运行在 RTX64 系统下的运动内核；（2）运动内核创建其与 EtherCAT 主站通讯的共享内存后，启动运行在 RTX64 系统下的 EtherCAT 主站，并进行参数初始化；（3）在运动内核中，根据事件流的执行顺序依次创建指令解析线程、连续轨迹预处理线程、插补计算线程、主站通讯线程、反馈线程，并根据线程优先级设定、扫描周期进行线程调度；（4）进入自动加工模式后，HMI 根据加工工艺需求将指令发送至指令共享内存。运动内核从指令共享内存中获取指令信息，并将轨迹规划信息发送至 EtherCAT 主站。同时，运动内核也从 EtherCAT 主站获取实时运动信息发送至反馈共享内存。HMI 根据反馈共享内存中的数据进行状态显示；（5）自动加工完成后，依次关闭运动内核、EtherCAT 主站、HMI，并完成必要的资源清理。

4　实验结果与分析

为验证基于 RTX64 的软运动控制器的性能，设计了如图 7 所示的硬件测试平台，该平台由占美 GK1037 工控机、高创 CDHD－0032A 系列高性能伺服驱动器与电机（共 12 组）、深圳市旗众智能科技有限公司的 I/O 从站与 box 从站

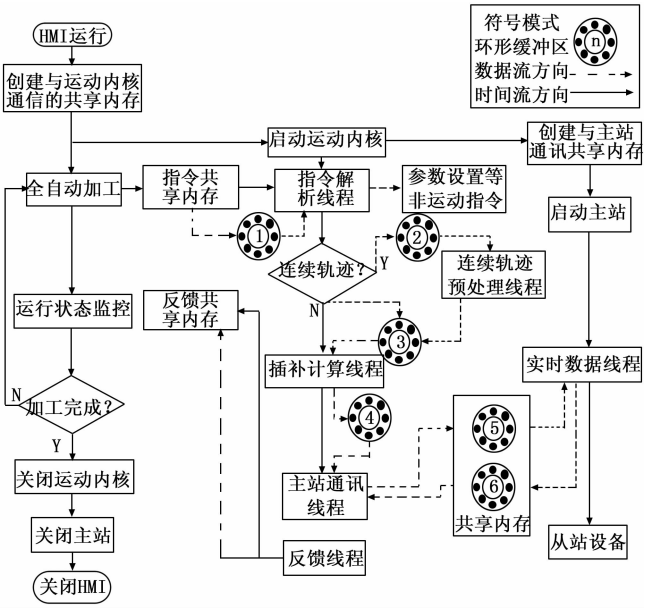

图 6　软运动控制器流程图

组成。图 8 为基于软运动控制器开发的手机外壳 3D 打磨系统的上位机界面。使用该上位机进行实验一与实验二的测试。

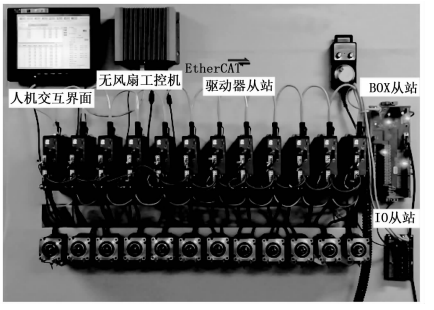

图 7　硬件测试平台

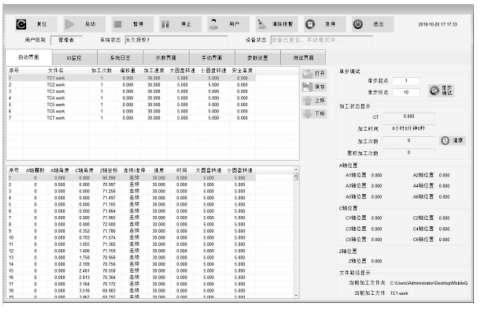

图 8　手机外壳 3D 打磨系统的上位机界面

实验一：使用 S 型算法进行单轴定长运动，位移长度为 6 000 pluse，起始速度、停止速度为 0 pluse/ms，目标速度为 50 pluse/ms，加速度、减速度为 1 pluse/ms^2，加加速度、减加速度为 0.01 pluse/ms^3。

每个通讯周期内的位置数据，如图 9 所示。可见位置曲线的变化较为平滑，且运动控制内核路径规划位置与伺

服驱动器反馈的实际运行位置偏差较小，均保持在两个周期的指令位置内，位置偏差控制在±0.01 mm 以内。表明软运动控制器具有较好的稳定性，较高的位置控制精度。

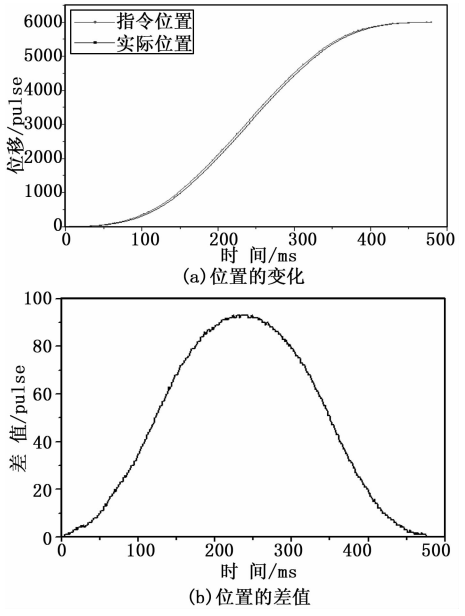

图 9 路径规划位置与实际运行位置信息

实验二：本实验通过在上位机的自动加工界面中编写磨削加工文件进行测试，实现手机外壳 3D 打磨系统的模拟效果。控制 XYZ 轴进行三轴联动插补，并以 X 轴、Y 轴为主轴，分别实现对 X0 至 X3 轴、Y0 至 Y3 轴的轴组控制；同时进行一维直线运动，即同时控制 12 个驱动器从站工作。

使用 RTX64 提供的 Tracealyzer 工具，可以监控实时子系统中多线程切换、调度时序等，有助于分析程序中的资源冲突、性能瓶颈，提高系统的实时响应，如图 10 所示。

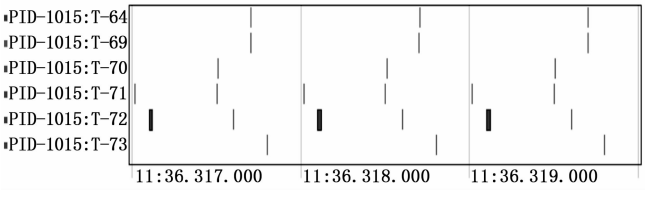

图 10 线程调度时序图

其中，PID－1015 表示运动内核当前进程 ID，T－64、T－69 至 T－73 分别表示主函数线程、上位机指令解析线程、连续轨迹预处理线程、插补计算线程、主站通讯线程、状态反馈线程的 ID。11：36.317.000 表示当前时间为 11 分 36 秒 317 毫秒 0 微秒。

由图 10 可知，各线程在运行态的时间均在设定的调度周期内，且均有 90％以上的时间冗余，证明软运动控制器具有较好的实时性与扩展性。

5 结束语

本文采用“Windows7＋RTX64”的设计方案，充分利用 Windows 系统资源丰富、界面友好的特性与 RTX 子系统强大的实时性能，设计并实现了开放性强、扩展性高的通用软运动控制器。通过软控制器控制所有的控制任务，并通过测试平台进行验证，实验结果表明：

1）基于 RTX64 的软运动控制器具有较高的稳定性与实时性，满足同步精度与位置误差精度的要求。

2）多线程调度时序符合设定的运动控制任务拓扑序列，满足了运动控制功能与实时控制性能的需求，具有较好的开放性与拓展性。

参考文献：

[1] 周 济. 智能制造——“中国制造 2025”的主攻方向 [J]. 中国机械工程，2015，26 (17)：2273－2284.

[2] 吴 宏，蒋仕龙，龚小云，等. 运动控制器的现状与发展 [J]. 制造技术与机床，2004 (1)：24－27.

[3] 陈日东. 基于 EtherCAT 的运动控制器研究与实现 [D]. 广州：华南理工大学，2017.

[4] Brecher C，Verl A，Lechler A，et al. Open control systems：State of the art [J]. Production Engineering. 2010，4 (2)：247－254.

[5] 朱达宇，李 彦，吉 华，等. 基于 RTLinux 的全软件数控系统 [J]. 计算机集成制造系统，2004 (12)：1571－1576.

[6] 王 普，张 蕾，郝立伟. 基于 RTX 的全软件数控系统的研究 [J]. 燕山大学学报，2007 (6)：513－516.

[7] Yu D，Hu Y，Xu X W，et al. An Open CNC System Based on Component Technology [J]. IEEE Transactions on Automation Science and Engineering，2009，6 (2)：302－310.

[8] 宋利利，王 宏. RTX 的三轴仿真转台实时控制软件设计与实现 [J]. 哈尔滨理工大学学报，2011，16 (3)：22－25.

[9] 毕鲁雁，刘立生. 基于 RTX 的工业机器人控制系统设计与实现 [J]. 组合机床与自动化加工技术，2013 (3)：87－89.

[10] Jiewen W，Di L，Shiyong W. The design and experimental research of an open architecture soft－CNC system based on RTX and an IPC [J]. The International Journal of Advanced Manufacturing Technology. 2017，89 (5)：1387－1399.

[11] RTX64 and Windows [EB/OL]. https://help.intervalzero.com/product_help/RTX64/RTX64_Help_Web.htm#PROJECTS/Subsystem/RTSS_Architecture/Adding_RTX_To_Win_Environment.htm，2018－10－24.

[12] 李忠民. 环形缓冲区的扩展方法 [J]. 电子技术与软件工程，2016 (4)：170－171.

[13] 徐勤朋，刘荫忠. 基于缓冲区的两级插补器的研究与设计 [J]. 组合机床与自动化加工技术，2014 (3)：9－12.

[14] 吴丽菲. EtherCAT 在实时系统下的实现 [D]. 广州：华南理工大学，2014.

[15] 张淑珍，于子然，孔民秀，等. 基于 RTX 弧焊机器人控制系统的设计与实现 [J]. 上海交通大学学报，2016 (10)：1540－1544.