

连接 CNC 系统和套料软件的中间件设计与实现

李科林, 王静静, 瞿少成

(华中师范大学 物理科学与技术学院电子信息工程系, 武汉 430079)

摘要: 针对目前工业自动化用户的个性化需求, 设计了一种连接 CNC 系统和套料软件的中间件系统; 首先设计了一种适合普通工程师理解与使用的人机界面, 并基于过程控制系统的 PVI 通信方式, 实现 CNC 系统数据的实时显示与跟踪; 通过 FTP 协议, 将标准套料软件生成的 G 代码传输到 CNC 系统, 并采用消息机制的进程通信方式, 实时回传 CNC 系统自动解析并执行 G 代码时的运行参数, 保证 CNC 系统执行路径与套料软件规划路径的一致性; 最后, 以某公司套料软件与 B&R 激光切割机 CNC 系统为对象, 设计并实现了该中间件; 经过工业现场测试表明, 该中间件能够连接到 B&R 公司目前所有的硬件系统上, 稳定性良好, 并能满足用户个性化的开发, 具有一定推广价值。

关键词: FTP; PVI 通信; 消息机制; 中间件

Design and Implementation of Middleware for Connecting CNC System and Nesting Software

Li Kelin, Wang Jingjing, Qu Shaocheng

(Department of Electronics and Information Engineering, College of Physical Science and Technology,
Central China Normal University, Wuhan 430079, China)

Abstract: Aiming at the personalized need of current industrial automation users, a middleware system is designed to connect CNC system and nesting software. Firstly, a man-machine interface suitable for ordinary engineers to understand and use is designed, and based on the PVI communication mode of process control system, the data of CNC system can be displayed and tracked in real time; The G code generated by the standard nesting software is transmitted to the CNC system through the FTP protocol, and the process communication mode of the message mechanism is adopted to send back the operation parameters of the CNC system when it automatically analyzes and executes the G code in real time, so as to ensure the consistency between the execution path of the CNC system and the programming path of the nesting software. Finally, the middleware was designed and implemented with a company's nesting software and the CNC system of B&R laser cutting machine. After industrial field tests, the middleware can connect to all the current hardware systems of B&R company with good stability and can meet the personalized development of users, which is worth promoting.

Keywords: FTP; PVI communication; message mechanism; middleware

0 引言

在现代工业生产过程中, 过程控制系统占据着重要的地位。近年来, 随着科学技术的发展, 过程控制自动化取得了长足的进步。然而, 随着用户需求的提高, 以及生产规模的扩大, 工控系统的功能在日趋强大的同时, 结构也愈发复杂, 目前已形成了多厂商软硬件产品共存的异构环境。在市场日趋激烈的竞争驱使下, 各自动化系统厂商迫切希望能够实现应用软件和工控设备的异构整合, 达到提升现场作业效率和企业个性化成本优化的效果^[1]。

作为一种高效、先进的工艺方法, 数控切割在工业制造

领域的应用愈加广泛。在实际的数控加工中, CNC 系统通过解释并执行 G 代码文件来实现工业的自动化加工。然而, 由于市场分工的精细化, 工业设计和现场加工往往是独立分工的, 所以 G 代码文件往往不是直接在 CNC 系统中生成的。G 代码的传输主要有手动输入、信息载体传输和 RS232 串口传输^[2]等方式。但是传统的传输方法存在速度慢、误差大、传输距离短等缺点^[3]。随着加工规模的扩大、插补精度的提高, 传统的传输方法已不能满足当下数控加工的需求, 用户需要一种更快速、准确的方法传输 G 代码^[4]。

为满足市场需求, 充分利用异构环境下 CNC 系统和套料软件各自的优势, 本文设计并实现了一款中间件系统。该中间件通过 PVI Services 接口实现上位机与贝加莱 CNC 系统的通信设计, 采用 FTP 文件传输协议将套料软件生成的 G 代码文件传输到 CNC 系统工件程序存储区, 然后通过 Windows 消息机制实现套料软件和中间件的进程通信。

1 中间件整体设计

中间件的整体功能设计如图 1。首先, 中间件作为 FTP 文件传输协议的客户端, 工控设备上的 CNC 系统作为 FTP

收稿日期: 2018-11-19; 修回日期: 2018-12-13。

基金项目: 国家自然科学基金(61673190); 中央高校探索创新项目(CCNU18TS042)。

作者简介: 李科林(1992-), 男, 广西玉林人, 硕士研究生, 主要从事深度学习与计算机智能排样方向的研究。

瞿少成(1971-), 男, 湖北荆州人, 教授, 博士生导师, 主要从事智能信息处理与非线性方向的研究。

文件传输协议的服务端，在中间件和 CNC 系统之间建立文件传输通道，将套料软件绘图、编程生成的 G 代码文件传输到 CNC 系统，从而实现了数控切割程序的高速远程双向传输。考虑到数控加工中实际的生产需要，并且为了充分体现中间件设计的实用性、灵活性以及高效性等特点，对于 CNC 系统中已有的加工文件，中间件还可以提供下载、删除等操作。

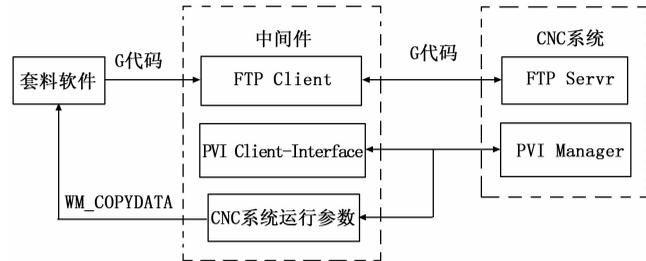


图 1 中间件整体功能设计

其次，中间件重写了 PVI 客户端接口，对 CNC 系统中的底层通信协议和数据类型进行封装，为应用层软件系统提供统一的调用接口，方便中间件系统的开发和维护升级。

最后，中间件通过封装的 PVI 通信协议与 CNC 系统建立通讯连接，实时获取和显示 CNC 系统的运行状态，并设计了一种便于普通工程师使用和操作的人机交互界面，从而实现对 CNC 系统状态和运行参数的控制。此外，中间件通过 WM_COPYDATA 消息与套料软件实现进程通信，实时传送 CNC 系统的切割状态到套料软件，以保证实际加工路径与规划路径的一致性。

2 通讯模块设计

2.1 PVI 通信模块

PVI (Process Visualization Interface) 是为了接入 B&R Automation Net 提供给所有 Windows 平台上软件程序包的通用接口^[5]。使用 PVI，既可以统一、协调的访问各种媒体和协议，也能在处理所有类型的数据的同时对设备进行控制。在开发应用程序时，不必花费时间和精力探索底层之间的通信，只需对 PVI 的对象结构进行简单配置，就能访问和操作控制器中的变量。

针对 PC 端 Windows 应用程序，B&R 提供了两类 PVI 客户端接口 (PVICOM interface 和 PVIServices components) 来实现应用程序与贝加莱 CNC 系统之间的 PVI 通信。PVICOM interface 基于 PviCom.dll，早期多应用于 Visual C/C++ 平台，且早期的 runtime 也仅支持 PVICOM 方式。随着 B&R 工控系统的发展和 runtime 环境的升级，且由于 PVICOM interface 缺乏实时的升级维护，目前在 PC 端应用程序开发的过程中已经很少作为客户端接口。

PVIServices components 客户端接口基于 BR.AN.PVIServices.dll，被所有基于 Windows 应用的 Visual Studio.NET 用于控制器的通信和诊断服务上。PVIServices 是基于 PVICOM 接口实现，并在编程环境中通过面向对象的结构呈

现，函数和变量数据都以 PVIServices Namespace 的形式，被整体分组在易于用户使用的类中，实现起来更加简单，并且 PVIServices components 升级维护良好，支持目前市场上几乎所有 B&R 系列的工控系统，其对象结构如图 2。因此，本中间件系统采用 PVIServices components 接口作为 PVI Client-interface 实现中间件与 CNC 系统的通讯。

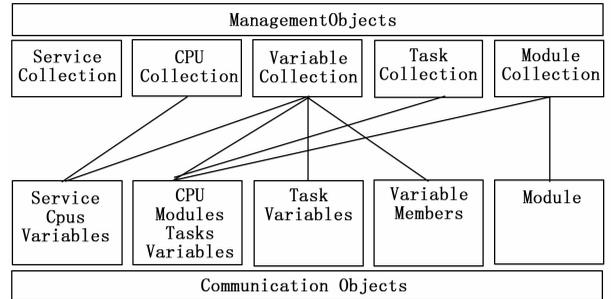


图 2 PVIServices 对象结构

2.2 G 代码文件的传输

计算机网络技术的发展加速了数控系统的网络化，为了满足当下数控加工的需求，通过网络快速高效传输数控加工程序，是网络化加工制造的必然趋势^[6]。FTP 是一个 8 位的客户端-服务器协议，可以操作任何类型的文件，是计算机网络中使用最广泛的文件传输协议，它屏蔽了计算机系统的诸多细节，非常适合在异构系统中实现文件的传输^[7]。因此，本文设计的中间件可以充分利用网络协议的开放性和通用性，使用基于 FTP 的文件传输协议，将套料软件生成的 G 代码文件传输到工控设备的 CNC 系统，实现快速高效的网络传输。

FTP 是一个客户机/服务器 (C/S) 系统，不同于其他基于 C/S 架构的系统，FTP 在客户端和服务端之间建立两条 TCP 连接来完成文件传输^[8]，一条连接负责传输控制信息 (命令和响应)，另外一条负责数据的传送，其结构模型如图 3。

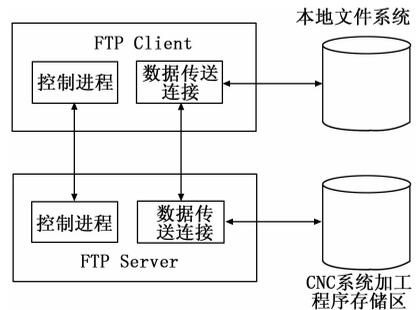


图 3 FTP 客户端/服务端模型

在本文设计的中间件中集成 FTP-Client 模块，通过与控制器上的 FTP Server 建立连接，就能将套料软件生成的本地 G 代码文件传输到 CNC 系统的加工程序存储区，有效解决了大容量加工程序的传输问题，既能满足当今数控系统的加工生产需求，又提高了数控加工程序传输的速度、可靠性。此外，中间件系统设计的客户端界面可以进行文

件的下载、删除和更新等操作, 非常便于工业现场的操作。

2.3 中间件与套料软件的进程通信

进程通信是指在进程间进行数据的传输(交换信息)。根据交换信息量的多少和效率的高低, 进程通信可以分为低级通信和高级通信。低级通信只能传递状态和整数(控制信息), 传送信息量小, 效率低, 每次通信传递的信息量固定。高级通信提高了信号通信的效率, 可以传递大量数据, 减轻程序编制的复杂度。其中高级进程通信主要分为三种方式: 共享内存模式、消息传递模式、共享文件模式。消息传递模式通过操作系统的相应系统调用进行消息传递通讯, 分为点到点发送的直接通讯和以信箱为媒介进行传递的间接通讯。消息传递的间接通信方式易于建立双向通讯链, 通讯安全且实现简单, 因此, 中间件与套料软件之间的进程通信可以通过基于消息传递的方式实现。

本文设计的中间件基于 Windows 系统平台, Windows 系统是一个由消息驱动的操作系统^[9], 而 Windows 消息机制是消息传递的间接通信方式中的一种, 相比于其他同机进程通信方式, Windows 消息机制实现较为简单、有效^[10]。

WM_COPYDATA 是一种功能强大又非常简单的 Windows 消息^[11], 且中间件与套料软件的通讯满足使用 WM_COPYDATA 消息的两个条件, 即知晓接收消息进程的句柄, 接收消息进程重载消息映射后可以给出反馈。为了实时回传 CNC 系统自动解析并执行 G 代码时的运行参数, 保证 CNC 系统执行路径与套料软件规划路径的一致性, 本文通过 WM_COPYDATA 消息机制实现套料软件与中间件不同进程之间的数据交互, 中间件作为消息发送方, 将要发送的信息打包, 并寻找作为接收方的套料软件的句柄, 找到接受窗体的句柄后将消息发送。当接收方接收到消息时, 先判断接收到的消息是否为 WM_COPYDATA 消息, 确认消息正确后才进行接收, 程序设计框图如图 4。

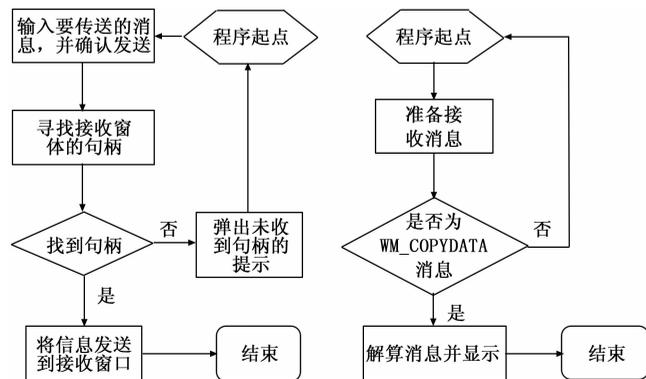


图 4 消息的发送与接收

3 中间件系统的实现

3.1 PVI Services 接口实现

中间件重写的 PVI Services 接口可以识别的通信对象有: 服务对象 (Service); CPU 对象 (CPU); 任务对象 (Task); 变量对象 (Variable) 和模型对象 (Module), 每个通信对象都由特性、事件与方法三种属性构成, 如图 5。

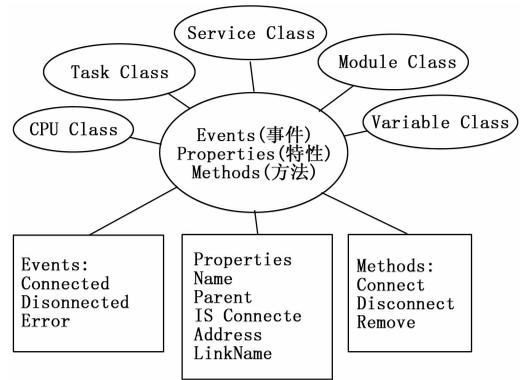


图 5 PVI Services 对象属性

其中, Service 对象负责创建连向 PVI Manager 的连接, 是 PVI 客户端中其他对象 (CPU, 任务, 变量等) 操作的基础, 当一个新的服务对象创建成功后, 即表示一个中间件与控制器系统的远程连接已经被建立。CPU 对象则负责建立 PVI Client 与控制器的连接。当 Service 对象和 CPU 对象创建连接成功后, 则代表过程变量的映像已经建立好, 即可对控制器 CNC 系统上的变量对象进行读写操作。其函数原型如下, 其中 cpu_name 设定为 ARsim 或 Remote_PLC, 相应的 ip_address 为默认 IP 和实际 PLC 的 IP 地址。

```
private void service_Connected(object sender, PviEventArgs e)
{
    myCpu = new Cpu(service, cpu_name);
    myCpu.Connected += new PviEventHandler(myCpu_Connected);
    myCpu.Connection.ANSLTCp.DestinationIpAddress = ip_address;
}
```

Task 对象与控制器的任务相对应, 管理任务中的全局和本地变量。Module 对象定义一个在控制器上的 BR 模块的属性, 可以对模块进行上传、下载和删除等操作。Variable 对象则表现为一个控制器上的变量, 可用变量类型有内在变量、全局变量、本地变量, 输入不同的高层次对象作为父类就可以创建不同类型的变量。将 Service 对象和 CPU 对象作为父类输入, 就分别创建了一个内在变量和一个全局变量。

3.2 WM_COPYDATA 实现

中间件使用 FindWindow 函数获得套料软件的句柄, 然后调用 SendMessage 函数向套料软件发送 WM_COPYDATA 消息, 其原型如下:

```
SendMessage(hwnd, WM_COPYDATA, wParam, lParam);
```

hwnd 表示接收窗体句柄, 发送消息类型为 WM_COPYDATA, wParam 表示发送窗口的句柄, lParam 为指向一个 COPYDATASTRUCT 结构的指针。COPYDATASTRUCT 结构声明如下:

```
typedef struct tagCOPYDATASTRUCT{
    DWORD dwData;
    DWORD cbData;
```

```
PVOID lpData;
} COPYDATASTRUCT, * PCOPYDATASTRUCT;
```

dwData 指定传递给接收方的数据, cbData 是要传递的数据块大小, lpData 是指向内存块的指针。中间件发送 WM_COPYDATA 消息时, 我们对 dwData 和 lpData 所传送的数据进行了自定义设置, 并且使用小端模型存储数据。

当 dwData=1 时, 设定当前发送的消息数据为单个变量, lpData 所指向的字节流定义如下:

变量 N 变量数据长度 N 变量数据 1 变量数据 2 变量数据 3...变量数据 N。

当 dwData=2 时, 表示消息发送的是多变量数据, lpData 指向的字节流如下:

变量 1 变量 1 数据长度 N 变量 1 数据 1 变量 1 数据 2 ... 变量 1 数据 N;

变量 2 变量 2 数据长度 N 变量 2 数据 1 变量 2 数据 2 ... 变量 2 数据 N;

...

变量 N 变量 N 数据长度 N 变量 N 数据 1 变量 N 数据 2 ... 变量 N 数据 N。

中间件发出 WM_COPYDATA 消息后, 套料软件作为接收方在消息映射表中添加 ON_WM_COPYDATA() 的映射, 并定义接收消息的响应函数:

```
OnCopyData(CWnd * pWnd, COPYDATASTRUCT * pCopyDataStruct);
```

套料软件通过中间件句柄 pWnd 向中间件发送反馈数据, pCopyDataStruct 完成对数据块的接收。

3.3 FTP 客户端实现

中间件所实现的 FTP 客户端具有易于操作与功能完善的图形界面, 包括用户登陆与退出、文件及目录的下载与上传, 方便工程师对 CNC 系统中的 G 代码文件进行下载、删除和更新等操作。

FTP 接口程序部分函数如下:

1) 连接 FTP 服务。

```
bool ConnectFtpServer (string strIp, UINT nPort, string strUserName, string strPassword, bool bPassive = TRUE, bool bUTF8Mode = TRUE);
```

其中: strIp 为 FTP 地址; nPort 为 FTP 服务端口, 默认端口为 21; strUserName 和 strPassword 为 FTP 用户名的密码; bPassive 表示是否为被动模式; bUTF8Mode 表示编码方式是否为 UTF8 编码。

2) 上传 FTP 文件。

```
bool UploadFtpFile (string strRemotePath, string strRemoteFile, string strLocalFilePathName);
```

其中: strRemotePath 为 FTP 相对目录; strRemoteFile 为 FTP 文件名称; strLocalFilePathName 则表示本地文件名称。

3) 删除 FTP 文件。

```
bool DeleteFTPFile (string strRemotePath, string strRemoteFile);
```

其中: strRemotePath 为要删除的 FTP 文件相对目录; strRemoteFile 为要删除的 FTP 文件名称。

3.4 数据交互与界面显示

中间件与 CNC 系统之间通过 PVI 通信协议、FTP 协议连通之后, 即可进行 CNC 系统状态等数据的传输以及各状态参数的显示与控制。

为了保证中间件系统的可读性, 便于中间件系统的升级维护, 中间件在进行界面变量信息的定义时, 遵照 CNC 系统的变量命名规则, 与 CNC 系统上的变量名称、类型保持一致。但是由于 CNC 系统变量结构复杂, 数量庞大, 为了保证变量映射关系的一一对应和升级维护的便利性, 中间件设计实现中通过交互变量的映射表构建变量配置文件, 通过配置文件的索引来关联变量参数, 这样既可以保证变量关联的准确性, 还可以便于交互变量的增加或减少, 设计灵活高效。

中间件可以接收 CNC 系统所传输的机床运行状态及配置参数, 并采用了一种简洁、易于理解和使用的图形界面来呈现 CNC 系统的机床状态、工艺参数等内容。中间件的操作界面在 CNC 系统界面显示的基础上, 充分考虑工程师的使用权限和操作习惯, 针对性的显示了切割控制、激光调整等各类控件, 对相应权限范围下不可操作的参数进行锁定, 可以降低现场因误操作导致的损失。

3.5 系统实现

为了保证中间件系统的兼容性和可扩展性, 本文设计的中间件以 Microsoft Visual Studio 2013 为开发环境, 采用 C++ 语言基于功能强大的 MFC 类库进行开发。中间件系统功能模块如图 6, 主要由通讯连接模块、数据交互模块和测试模块三个部分组成。

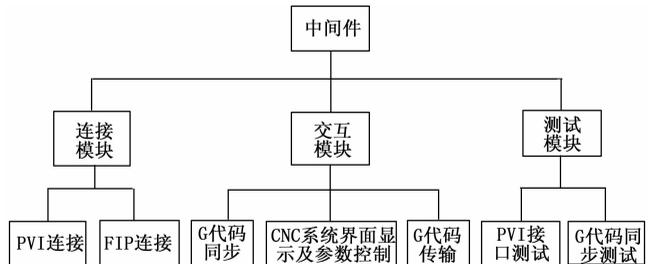


图 6 中间件功能模块

通讯连接模块包含了中间件与 CNC 系统的 PVI 连接和 FTP 连接设置。图 7 是中间件连接界面, 用户可以根据需求配置中间件与 CNC 系统的通讯连接。PLC 类型选择 AR-Sim 时, 默认 IP 地址 127.0.0.1, 即实现连接虚拟的 PLC 进行仿真, 便于在升级维护的开发过程中进行调试; 在实际的应用现场中, PLC 类型统一设定为 Remote_PLC, 需要输入实际 PLC 的 IP 地址。在 FTP 服务设置中, IP 地址为实际连接的 PLC 地址, 端口默认 21, 用户名和密码分别对应 CNC 系统设置的 FTP 账号密码, 相对路径为 G 代码在 CNC 系统的存储路径, 默认为 /DATA/Programs, 实际操作可以根据具体需求进行设定。



图 7 CNC 连接设置

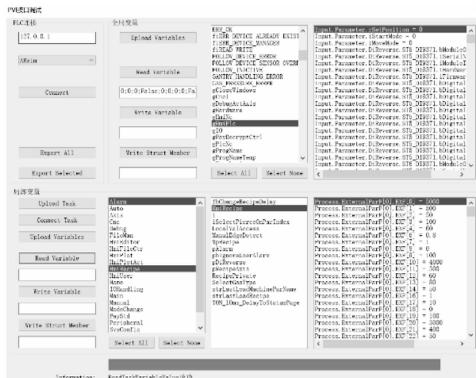


图 9 PVI 接口测试

中间件的数据交互模块实现了 CNC 系统的机床状态、标定调整、工艺参数、机床维护等 16 个界面以及 G 代码的传输和同步页面。中间件与 CNC 系统连接成功后, 就可以实现 CNC 系统运行状况的实时显示和控制。本文设计的中间件主要采用的是 B&R X20 系列控制器进行实验测试, 图 8 是中间件与 CNC 系统连接成功后的激光调整界面, 界面上显示的是 CNC 系统当前的运行参数, 在界面中对相应状态参数进行修改, 点击应用保存, 就可以控制 CNC 系统的运行状况, 操作简单方便。



图 8 激光调整

系统数据的实时显示与跟踪, 保证了 CNC 系统执行路径与套料软件规划路径的一致性。系统实际应用表明, 该系统人机交互良好, 且实现了大容量数控程序的网络化高速传输, 能有效解决用户的个性化需求, 节约了企业生产成本, 具有一定的推广价值。

参考文献:

[1] 陆小虎. 开放式数控系统网络化平台关键技术研究与应用 [D]. 沈阳: 中国科学院研究生院 (沈阳计算技术研究所), 2015.

[2] 杨德斌. 数控机床的通讯及其网络的开发与应用 [J]. 制造技术与机床, 2003 (5): 39-42.

[3] 伍文进. 基于串口服务器的数控机床通讯的探索与研究 [J]. 制造技术与机床, 2015 (5): 137-141.

[4] 龙 虎. 数控机床的数据传输 [J]. 机械管理开发, 2018 (5): 151-152.

[5] 林国营, 赵继敏. 过程控制系统中基于 PVI 的通信设计 [J]. 工业控制计算机, 2007 (8): 24-25.

[6] 王 谦. 开放式数控系统的网络化实现研究 [J]. 化工设计通讯, 2018, 44 (6).

[7] 顾辰辰. FTP 客户端及服务器系统的设计与实现 [D]. 南京: 南京邮电大学, 2016.

[8] 刘 峰. 基于物联网的公路网传感设备故障信息系统设计 [J]. 计算机测量与控制, 2015, 23 (3): 766-769.

[9] 苏玉强, 刘 颖. 基于消息的嵌入式系统应用框架研究 [J]. 计算机测量与控制, 2011, 19 (5): 1142-1144.

[10] 王 亚, 宋铭利. Windows 消息机制研究 [J]. 现代计算机, 2008 (2): 70-71.

[11] 凌大鹏, 陆 平, 李 芳, 等. 基于 Win32 API 进程通信的方法研究 [J]. 舰船防化, 2008 (6): 48-52.

4 结束语

本文针对目前工业自动化用户的个性化需求, 通过整合套料软件和 CNC 系统的部分功能, 研究并开发了一种便于普通工程师理解和操作的中间件系统。通过将标准套料软件生成的 G 代码传输到并解析到 CNC 系统, 实现了 CNC

(上接第 230 页)

[12] Adams H. Asynchronous operations and Web services, Part 2 [J]. IBM Systems Journal, available on ibm.com/developer, 2002.

[13] 崔 伟, 汪诗林. 分布式系统中数据同步机制的研究与实现 [J]. 计算机工程与设计, 2007, 28 (10): 2259-2261.

[14] 段 弘. 基于 Play 的用户匹配与内容推荐系统设计与实现 [D]. 成都: 电子科技大学, 2013.

[15] 沈 雁. 基于消息队列的分布式图像处理系统的设计与实现 [D]. 杭州: 浙江大学, 2013.

[16] 唐 伟. 在 Windows 下实现配置 FTP 服务器 [J]. 计算机时代, 2004 (5): 41-43.