

[新设备·新材料·新方法]

DOI:10.3969/j.issn.1005-2895.2017.05.014

基于 WPF 的 OPC 技术在数控机床 管控系统中的应用

沈荣成^{1,2}, 张秋菊^{1,2}, 李克修¹, 左文娟¹

(1. 江南大学 机械工程学院, 江苏 无锡 214122;
2. 江南大学 江苏省食品先进制造装备与技术重点实验室, 江苏 无锡 214122)

摘 要:针对加工车间数控机床设备的实时监测与控制,考虑系统的可扩展性,采用专用和通用相结合的 OPC 通信技术,提高系统的扩展性。应用 .NET 平台下 WPF 框架,使用 C# 语言和 SQL 数据库技术开发和实现了数控机床管控系统。系统构建了基于 OPC Server 和多个 PLC 的管控系统,实现了 OPC Server 的通信和上位机软件的编程,完成了数据的采集、数据的处理、分析、信息挖掘、实时反馈和实时决策等功能。通过实际运行表明本管控系统工作稳定可靠,具有较大的推广价值。

关 键 词:数控机床;管控系统;OPC 技术;WPF 框架

中图分类号:TS152.7;TP319 文献标志码:A 文章编号:1005-2895(2017)05-0061-05

Application of OPC for Monitoring and Controlling System of CNC Based on WPF

SHEN Rongcheng^{1,2}, ZHANG Qiuju^{1,2}, LI Kexiu¹, ZUO Wenjuan¹

(1. School of Mechanical Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China;

2. Jiangsu Key Laboratory of Advanced Food Manufacturing Equipment & Technology, Jiangnan University, Wuxi, Jiangsu 214122, China)

Abstract: Aiming at the real-time monitor and control of CNC, considering the expand-ability of system, adopted the combination of special and general OPC communication technology to improve the augmentability of system, applied WPF framework in .NET platform and used C# language and SQL database to develop the monitoring and controlling system (MCS). MCS based on OPC Server and several PLC realized the communication among PLC (CPU SR20), OPC Server, PC software, completed the functions of data collection, data management, data analysis, information mining, real-time feedback and real-time decision-making. Through the actual operation shows that MCS is stable, reliable and has a great popularization value.

Keywords: CNC (numerical control machine); MCS (management and control system); OPC (object linking and embedding (OLE) for process control); WPF (windows presentation foundation)

在工业控制领域,随着通信和计算机技术的发展,企业对生产过程不断提出新的要求,特别是在人机交互界面及实时响应方面^[1-3]。在如今工业控制现场,大多数企业采用 PLC 作为下位机控制器实现生产过程中的自动化控制。上位机管控系统需要对工业运行流程监视及相应控制,则 PLC 与 PC 的相互通信是上层

管控系统实现的关键。OPC 技术是微软公司针对过程控制方面的应用而开发的^[4],基于 OLE/COM/DCOM 技术,采用 C/S 模式开发,为工业自动化软件面向对象的开发提供了标准。标准定义了应用 Microsoft 操作系统在基于 PC 的客户机之间交换自动化实时数据的方法。采用这项标准后,可快速方便地

收稿日期:2017-02-20;修回日期:2017-07-20

基金项目:江苏省科技计划项目(BE2015051)。

第一作者简介:沈荣成(1991),男,江苏盐城人,硕士研究生,主要研究方向过程监控及数控设备联网管控系统开发。E-mail: 1028127289@qq.com

开发出适合自己硬件的一套 OPC 客户端接口程序^[5]。随着 OPC 技术的不断发展,才使得 PC 与 PLC 之间的管控一体化变得简单高效^[6-8]。目前,厂商提供的一般都是专用的 OPC 服务器^[9],只能与公司自己的产品通信,降低了系统的可扩展性。也有一些公司提供通用 OPC 服务器,可以兼容很多产品,但是没有专用 OPC 服务器完整的通信支持。笔者综合使用通用型与专用型的 OPC 服务器,应用 .Net 平台下 WPF 技术带来的优秀视觉体验创建客户端应用程序^[10],并结合其界面开发和业务逻辑分离的新型 MVVM 框架开发数控机床管控系统^[11]。系统攻克了原先管控系统的实时性差,布线难,稳定性差等问题,并在系统的集成性、可扩展性、通用性等方面对管控系统进行了优化。

1 管控系统的实现

1.1 管控系统的构成

数控机床管控系统由上位管控系统控制机、OPC 服务器、下层采集终端这几个部分组成,系统架构如图 1 所示。

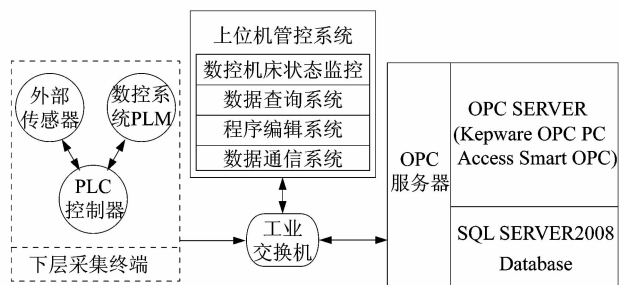


图 1 数控管控系统架构图

Figure 1 MCS architecture

系统通过工业交换机实现与各部分的通信,实时读取与写入系统运行参数,例如系统的外部控制信号、机床加工工艺参数、系统实时状态及系统数据管理等。系统下层采集终端为 S7-200 Smart PLC 和信捷 XC3 系列 PLC 混合控制器,通过串口服务器与工业交换机连接,可实现以太网无线与有线的混合通信。

1.2 专用与通用 OPC 服务器的组态

系统是基于 .NET 平台 WPF 框架,使用 C# 编程语言设计开发的,采用面向对象的编程方法,利用数据与控件绑定的方式实现数据通信。系统采用 PC Access Smart OPC 和 KepWare OPC 混合 OPC 服务器,通过系统对接入 PLC 的 CPU 的区分,实现 OPC 服务器的选择,保证系统最优化运行。

系统上位机服务器组态软件有:S7-200 PC Access Smart, STEP7-MicroWIN Smart V2.1, KEPWareServer

V4.5。通过将 IP 地址设置到同一号段,使用 OPC 软件的测试客户端对数据通信进行测试,能监测出 PLC 中数据则表示数据通信正常,OPC 服务器的组态完成。

1.3 OPC 客户端通信的实现

在上位机上已安装 OPC 服务器后,OPC 客户端应用程序通过连接服务器,就可以使用 OPC 对象支持的属性与方法,可对 PLC 中的数据实现读写操作。OPC 客户端实现步骤:①通过 IP 地址连接 OPC 服务器;②创建组和项并定义刷新周期;③遍历服务器点;④通过 OPC 接口以同步或异步的方式读取或写入数据;⑤将读取或写入的数据根据系统的需要,保存值数据库,并设标志位让数据库中的数据进行增、删、改及查的操作;⑥断开 OPC 客户端与 OPC 服务器之间的连接。下面具体介绍在 C# 环境下开发 OPC 客户端的流程。

1.3.1 添加库引用

新建 WPF 客户端,在 WPF 项目解决方案资源管理器下引用栏添加 OPC. Commom, OPC. Data 和 OPCAutomation 库。这类库是西门子公司和 KepWare 公司提供的支持 .NET 平台的类库。首先在程序的头文件中使用命名空间如下:

```
Using OpcAutomation;
Using OPC. Commom;
Using OPC. Data;
Using System. Windows. Threading;
Using System. Data;
Using System. Runtime. InteropServices;
```

1.3.2 OPC 服务器选择

系统在针对西门子 PLC 时,选用 PC Access OPC,保证系统与硬件的最大兼容。在系统中组态多种类型的 PLC 时,选用 KepWare OPC,使系统具有较大兼容性,保证系统能正常运行。下面定义了系统选择 OPC 服务器的句柄:

```
private void btn_Opc_Connect_Click(object sender,
RoutedEventArgs e)
{
    ComboBoxItem SelectedItem = Opc_server_
select.SelectedItem as ComboBoxItem;
    if (SelectedItem.Content.ToString() ==
"PC ACCESS SMART OPC")
    {
        Using_PCAccess_OPC();
    }
}
```

```

else
{
    Using_KepWare_OPC();
}
}

```

1.3.3 数据通信

OPC 服务器支持 2 种数据项读取类型,同步读取数据和异步读取数据。同步读取指在同步读取过程中,OPC 客户端程序停止执行其他指令,直到服务器读取数据为止,适应数据访问很快发生或需要为程序流提供结果的情况;异步读取指 OPC 客户端程序在 OPC 服务器读取数据时继续执行,适应程序中有不可接受的延迟的情况。由于本系统中有较大量的数据交互,实时性要求比较高,所以使用异步方式的效率更高。系统中的 2 类 OPC 服务器的客户端设计程序基本类似,以 PC Access Smart 为例介绍 OPC 客户端的设计。

在命名空间开头,先定义客户端全局变量:

```

OpcServer TheSrv;//新建 OPC 服务器
IOPCGroup TheGrp;//新建 TheGrp 组对象
OPCItemDef [ ] ItemDefs = new OPCItemDef
[20];//新建一个数组,接受 Item 项
Int [ ] HandlesSrv = new int [20];
OPCItemResult [ ] rltm = new OPCItemResult
[100];//OPC 服务器子项结果数组
string ServerProgID = " S7200SMART.
OPCServer";//OPC 服务器名称

```

1.3.4 创建服务器的连接与断开

新建一个 OpcServer 接口,通过该接口调用 Connect(ServerProgID)方法连接 OPC 服务器。

```

TheSrv = New OpcServer
TheSrv. Connect( ServerProgID)

```

在使用完 OPC 服务器后,系统可以通过调用 Disconnect()方法,依次删除 Item 和 Group,释放 OPC 服务器资源。

1.3.5 添加 Group 对象和 Item 对象

添加组对象需要调用 AddGroup()方法

```

TheGrp = TheSrv. AddGroup ( string groupName,
bool setActive, int requestedUpdateRate);

```

其中 groupName 为组名, setActive 为激活的标志, requestedUpdateRate 为数据的刷新频率,以 ms 为单位,最小值为 50 ms。

添加 Item 对象需要调用 AddItems()方法

```

For(int i=0;i < ItemsNo. Length;i + + )
{
    ItemDefs[ i ] = new OPCItemDef ( string id, bool
activ, int hclt, VarEnum vt);
}

TheGrp. AddItems( ItemDefs, rltm);

```

For 循环中的项的定义参数来源从 SQL2008 数据库中读取,并通过记录逐条加载。通过 id 号对每一 Item 子项进行区分。

在本系统中存在多个客户端,所以为了避免引起冲突,要求始终只有一个客户端处在被激活状态,这需要调用 SetClientName()方法:

```

public void SetClientName( string name);

```

1.3.6 读写操作及订阅方式下数据改变处理事件 同步读数据方法:

```

SyncRead( short Source, int NumItems, ref Array
ServerHandles, out Array Values, out Array Errors, out
object Qualities, out object TimeStamps);

```

异步读数据方法:

```

AsyncRead ( int NumItems, ref Array
ServerHandles, out Array Errors, int TransactionID, out
int CancelID);

```

同步写数据方法:

```

SyncWrite( int NumItems, ref Array ServerHandles,
ref Array Values, out Array Errors);

```

异步写数据方法:

```

AsyncWrite ( int NumItems, ref Array
ServerHandles, ref Array Values, out Array Errors, int
TransactionID, out int CancelID);

```

对于需要经常更新以显示数据项的部分,如管控系统的传感器数据值,通常使用“DataChanged”事件实现服务器到客户端的异步缓存数据传输。这样减少了客户端与服务器之间传输的数据,并允许服务器优化从设备采集的数据。“OnDataChange”数据处理方法:

```

TheGrp. DataChanged + = new System.
EventHandler( this. theGrp_DataChange);

public void theGrp _ DataChange ( object source,
DataChangeEventArgs e) {}

```

2 OPC 客户端软件测试

基于设计步骤,在 WPF 环境下开发下开发了 1 个 OPC 测试客户端。打开 OPC 测试界面,选择好 OPC 服务器,成功添加 OPC 组和项后,点击连接按钮,提示连接成功后,可以根据需要选择读写类型,实现与 PLC

的通信,通过上位机设置对 PLC 输出点的控制,可直观了解通信状况,从而实现管控系统与下层采集系统的通信。OPC 测试客户端界面如图 2 所示。



图 2 OPC 测试客户端界面

Figure 2 OPC test client interface

3 多种 OPC 在数控机床管控系统的应用

数控机床的管控系统是一对多的控制系统,所以针对的设备可以是一个品牌的,也可能是多个品牌的。针对这一情况,系统集成了通用和专用混合 OPC 服务器,使得任何一种设备的接入,只需要提供驱动就可以顺利接入系统,容易实现系统的扩展。

本管控系统已在某加工工厂得到实际应用,用于监控 5 台数控加工中心,底层采集端使用西门子和信捷两种品牌 PLC,控制器上使用 Kepware OPC 和 PC Access Smart OPC 用于提供 OPC 服务器功能。在管控系统中集成 OPC 客户端,将采集的数据通过 C# 定义的属性实时显示与保存至数据库中。系统中所监控的部分 Item 项如图 3 所示。

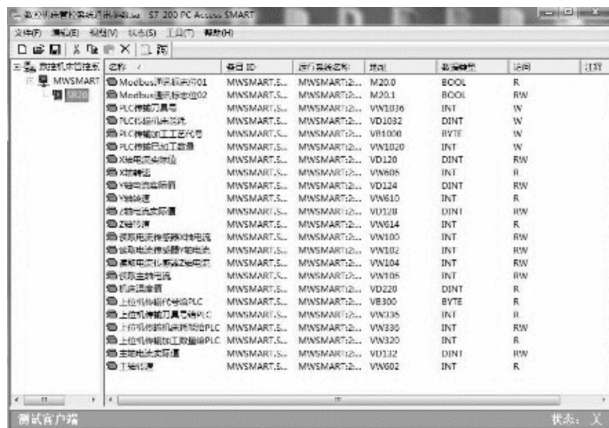


图 3 管控系统监控部分 Item 项

Figure 3 MCS monitor part of Items

应用结果表明,该系统可以实时对多台设备进行数据的读写,系统通过对数据阈值的给定,可以实时地做出对应的响应。另外,本系统可以为用户分析每台机床的运行状况,提供各个参数的图表显示功能,可通过观察实时曲线查看电机转速变化,各电机轴的电流变化,加工温度的变化。实现了机床能耗分析、零件加工工时统计、设备使用效率分析和机床能耗等柱状图显示及各类报表的输出。管控系统运行主界面如图 4 所示。

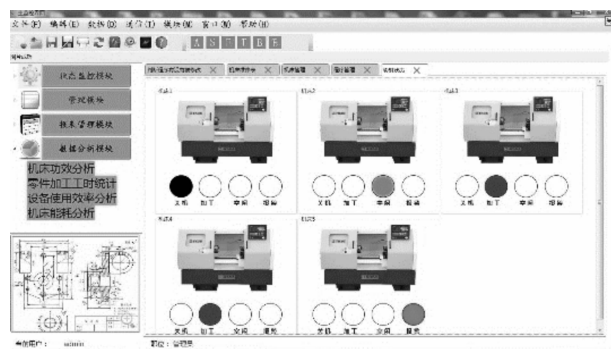


图 4 数控机床管控系统主界面

Figure 4 Main interface of MCS

4 结语

系统通过多种 OPC 技术、SQL2008 及 .NET 平台下 WPF 技术的结合,完成了上位机与底层数据采集端的数据通信、数据处理、信息保存、数据挖掘和实时决策等功能。系统中通过多种 OPC 服务器的集成,大大提高了系统的扩展性能,可作为其他工业管控系统的样例。开发出的数控机床管控系统已通过实际测试,系统运行稳定可靠、实时性好、可靠性高,具有一定的借鉴意义。

参考文献:

- [1] 陈铁军,李华伟. 基于 OPC 的化纤长丝卷绕设备监控系统设计与实现[J]. 郑州大学学报(工学版),2014,35(3):69-72.
- [2] 吴晗,杨颖,周亚军. 基于 OPC 的子午线轮胎生产过程监控系统[J]. 机电工程,2014,31(11):1509-1512.
- [3] 张宏坡,康龙云. 基于 OPC 技术的 PLC 上位机管理系统[J]. 制造业自动化,2011,33(18):72-75.
- [4] AHIN C, BOLAT E D. Development of remote control and monitoring of web-based distributed OPC system[J]. Computer standards & interfaces, 2009, 31(5):984-993.
- [5] 日本 OPC 基金会. OPC 应用程序入门[G]. OPC(中国)促进委员会,译. 北京:OPC 中国基金会,2002.
- [6] 李革,王忠民. 基于 XML Web Services 的管控一体化解决方案的研究[J]. 微计算机信息,2005,21(17):6-8.

(下转第 69 页)