

OPC 服务器开发工具包软件的设计与实现

王平, 李大庆, 王颀

(重庆邮电大学网络化控制与智能仪器仪表教育部重点实验室, 重庆 400065)

摘要: 针对源码级的用于过程控制的对象链接和嵌入(OPC)服务器开发过于复杂问题, 提出 OPC 服务器开发工具包软件。该软件以 Visual C++6.0 为开发平台, 通过将 OPC 数据存取规范和 COM 知识封装成 DLL 的方式实现。介绍该软件的结构和主要模块的功能, 设计 OPC 服务器。测试结果证明, 该软件设计提高程序代码的重用性, 缩短软件开发周期, 该服务器能实现基本的数据通信。

关键词: 用于过程控制的对象链接和嵌入; 组件对象模型; 开发工具包

Design and Implement of OPC Server Development Toolkit Software

WANG Ping, LI Da-qing, WANG Ting

(Key Laboratory of Network Control & Intelligent Instrument, Ministry of Education, Chongqing University of Posts & Telecommunications, Chongqing 400065)

【Abstract】 This paper proposes OLE for Process Control(OPC) server development toolkit software, aiming at exploitation of source OPC server too complex. Visual C++6.0 as the development platform, it encapsulates OPC data access criterion and COM knowledge to a single DLL to realize the software, and introduces the structure and primary module function of OPC server develop toolkit. Test result proves that the software design enhances reusability of code integer and reduces cycle of software development, the server can realize basal data communication.

【Key words】 OLE for Process Control(OPC); Componet Object Model(COM); development toolkit

目前, 在工控领域的 OPC(OLE for Process Control, 用于过程控制的对象链接和嵌入)服务器开发中, 由于源码级开发 OPC 服务器除了需要开发者掌握复杂的组件对象模型(Componet Object Model, COM)知识和 OPC 规范外, 还需要开发者拥有扎实的编程基础, 因此开发难度很大。在该情况下采用 OPC 服务器开发工具包软件可以简化 OPC 服务器的开发过程, 降低 OPC 服务器开发的难度^[1-2]。用户无须了解 COM 知识和 OPC 规范, 只须对 OPC 服务器开发工具包中对应的函数进行调用^[3]实现与底层设备通信的协议驱动程序, 就可以编写出相应的 OPC 服务器。

1 OPC 服务器开发工具包软件的体系结构

该工具包软件由 3 个部分组成: (1)动态链接库 OPC_CQUPT_Svr.dll; (2)头文件 OPC_CQUPT_SvrAPI.h; (3)静态库 OPC_CQUPT_Svr.lib。OPC_CQUPT_Svr.lib 静态库包含 DLL 中 API 函数的导出定义。在 OPC 服务器工程区文件中包含 OPC_CQUPT_Svr.lib 文件, OPC_CQUPT_SvrAPI.h 文件和 OPC_CQUPT_Svr.dll 文件, 通过此类模块对 DLL 进行调用。

该工具包软件的编程接口按功能划分为 3 类: 数据与属性管理, 实时数据管理和服务器管理。

(1)OPC 数据与属性管理: 组织 OPC 数据服务器的浏览地址空间, 提供 OPC 数据的属性管理。该类包含辅助功能函数、OPC 项函数等功能块。

(2)OPC 实时数据管理: 提供 OPC 服务器的实时数据读写功能。该类包含实时信息函数、调用返回定义等功能块。

(3)OPC 服务器管理: 搭建整个服务器框架, 提供 OPC 服务器的启动/停止、注册等功能。该类包含 OPC 框架功能块、初始化和注册函数功能块等。

OPC 服务器开发工具包软件设计框图如图 1 所示。

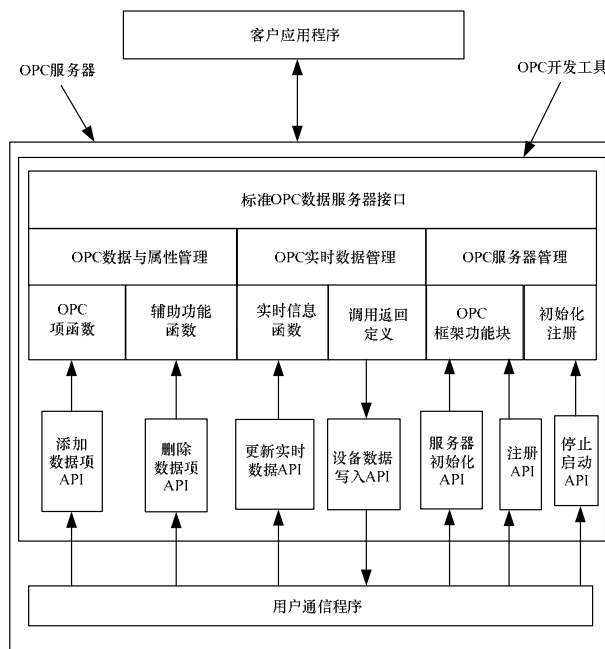


图 1 OPC 服务器开发工具包软件设计框图

基金项目: 国家“863”计划基金资助项目“EPA 网络监测分析软件”(2006AA040301-07)

作者简介: 王平(1963-), 男, 教授、博士生导师, 主研方向: 工业以太网及网络控制技术, 无线控制网络及应用, 智能仪器仪表; 李大庆, 硕士研究生; 王颀, 副教授

收稿日期: 2009-05-21 **E-mail:** qing_0906@163.com

2 OPC 服务器开发工具包软件各模块设计

OPC 服务器开发工具包软件动态链接库 OPC_CQUPT_Svr.dll 主要包含 3 大类, 6 大功能模块。

(1) OPC 框架模块

该模块负责创建所有 OPC 规范要实现的 COM 对象, 如 OPC 服务器接口对象, OPC 组接口对象, 以及实现 OPC 规范规定的全部功能。该模块实现的主要功能如下:

- 1) 完成 OPCServer COM 对象的创建, 用 OPCServerlist 来维护 OPC 服务器链表;
- 2) 完成 OPCGroup COM 对象的创建, 用 GroupList 来维护 OPC 组链表;
- 3) 完成对 Tag 的添加、删除、刷新、读操作、写操作等功能;
- 4) 实现数据的同步和异步访问方式;
- 5) 维护一张 Tag 名字的 Hash 树, 协助 CIOPCBrowse 实现浏览 Tag。

(2) 调用返回定义模块 API 函数定义

该模块明确定义了 DLL 用于返回给服务器程序调用返回的具体含义, 以实现客户应用与服务器应用之间的数据交换, DLL 提供以下 5 个 API 函数:

BOOL EnableUnknownItemNotification(UNKNOWNITEMPROC lpCallback)

函数功能: 该函数的返回值表示一个客户请求一个没有定义或是已经被暂停刷新的标签, 但无论客户请求的标签是否是已定义的标签, 该函数都定义一个从 DLL 返回的调用。

BOOL EnableItemRemovalNotification(ITEMREMOVEDPROC lpCallback)

函数功能: 当对一个标签的最后客户引用释放后, 该函数将定义一个来自 DLL 的调用返回, 服务器应用程序可以删除该标签或暂停该标签的刷新。

BOOL EnableDisconnectNotification(DISCONNECTPROC lpCallback)

函数功能: 当客户应用请求断开连接时, 该函数产生 1 个来自 DLL 的调用返回通知服务器应用断开连接。

BOOL EnableWriteNotification(WRITENOTIFYPROC lpCallback, BOOL ConvertToNativeType)

函数功能: 当 OPC 客户写一个已定义的标签时, 该函数产生一个来自 DLL 的调用返回。

BOOL EnableDeviceRead(DEVICEREADPROC lpCallback)

函数功能: 该函数允许应用程序提供一个回调线程来处理来自于客户端的设备读请求。

(3) 初始化与注册函数模块 API 定义

该模块实现 OPC 服务器的注册和注销, 以及更新注册的操作。DLL 提供以下 3 个 API 函数:

BOOL InitOPCSvr(BYTE *pCLSID_Svr, UINT ServerRate)

函数功能: 该函数实现对 DCOM 的初始化和创建 OPC 服务器对象。

BOOL UpdateRegistry(BYTE *pCLSID_Svr, LPCSTR Name, LPCSTR Descr, LPCSTR ExePath)

函数功能: 该函数封装 Windows 注册函数, 用于 OPC 服务器的注册。

BOOL UnregisterServer(BYTE *pCLSID_Svr, LPCSTR Name)

函数功能: 这个函数是服务器注销函数, 根据 OPC 服务器的类标识(pCLSID_Svr)和服务器名(Name)注销相对应的 OPC 服务器。

(4) OPC 项函数 API 定义

函数功能: 它是 1 个服务器注销函数, 其参数 pCLSID_Svr 是 OPC 服务器的类标识(CLSID); 参数 Name 为服务器的名字。该模块完成开发工具包的最核心操作: 标签创建, 标签更新和标签移除等操作。DLL 提供以下 6 个 API 函数:

HANDLE CreateTag(LPCSTR Name, VARIANT Value, WORD InitialQuality, BOOL IsWriteable)

函数功能: 该函数实现在动态链接库中创建过程标签。如果标签创建成功, 那么函数返回数据点的句柄(HANDLE), 否则返回 INVALID_HANDLE_VALUE。

BOOL UpdateTag(HANDLE TagHandle, VARIANT Value, WORD Quality)

函数功能: 该函数实现过程标签的数据刷新。

BOOL UpdateTagWithTimeStamp(HANDLE TagHandle, VARIANT Value, WORD Quality, FILETIME timestamp)

函数功能: 该函数允许应用改变一个已定义的标签值、标签质量和标签时间。

BOOL SetTagProperties(HANDLE TagHandle, DWORD PropertyID, LPCSTR Description, VARIANT Value)

函数功能: 任何数量的 OPC 项属性都可以用该函数设置, DLL 保持了所有已定义属性的一个列表, OPC 客户可以访问此类属性。

BOOL ReadTagWithTimeStamp(HANDLE TagHandle, VARIANT *pValue, WORD *pQuality, FILETIME *pTimestamp)

函数功能: 应用程序使用该函数去获取一个标签的当前值。

BOOL RemoveTag(HANDLE TagHandle)

函数功能: 该函数从 DLL 中删除一个过程标签。

(5) 实时信息 API 定义

该模块提供了一次刷新多个标签的方法。DLL 提供以下 3 个 API 函数:

BOOL StartUpdateTags()

函数功能: 开始刷新。

BOOL WINAPI UpdateTagToList(HANDLE TagHandle, VARIANT Value, WORD Quality)

函数功能: 刷新标签列表。

BOOL WINAPI EndUpdateTags()

函数功能: 结束刷新。

上述函数必须一起使用, 才能提供一次刷新多个标签的功能。首先调用 StartUpdateTags, 然后用 UpdateTagToList 代替 UpdateTag()调用, 当所有标签都被刷新后, 再调用 EndUpdateTags()结束此次刷新操作。

(6) 辅助功能函数 API 定义

该模块为 OPC 服务器提供一些辅助操作。DLL 提供以下 4 个 API 函数:

BOOL RefreshAllClients()

函数功能: 服务器可以使用该函数对所有连接的客户进行初始化。

BOOL ResetServerRate(UINT ServerRate)

函数功能: 该函数用于改变服务器应用的数据刷新率。

BOOL SetVendorInfo(LPCSTR VendorInfoString)

函数功能: 该函数设置一些服务器的版本信息。

void RequestDisconnect()

函数功能: 该函数提供了请求所有客户断开连接的功能。

各功能模块间的关系如图 2 所示。

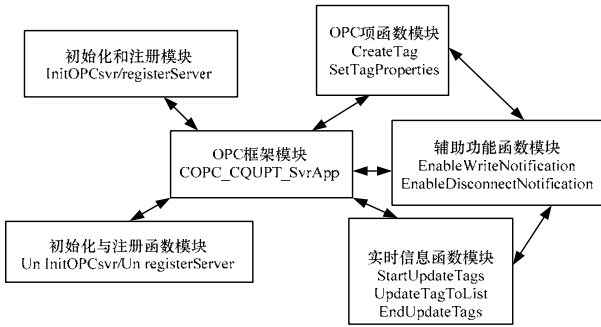


图 2 OPC 服务器开发工具包软件各模块间的关系

3 OPC 服务器

OPC 服务器由 3 个功能模块组成：OPC 标准接口模块，服务器界面模块和硬件设备通信的专用通信协议模块。3 个模块通过同一块内存数据区来共享数据，通过线程的同步和互斥等技术，解决共享数据的保护问题。各模块功能如下：

- (1)OPC 标准接口模块：实现 OPC 数据存取规范规定的几乎所有功能。本文将通用模块做成 DLL 的形式，供用户调用，可以很方便地实现该模块^[3-4]。
- (2)服务器界面模块：该模块主要完成各标签的添加、删除、注册等操作。
- (3)专用通信协议模块：该模块是一个针对硬件设备的驱动程序。在此采用基于 EPA 现场总线通信协议的驱动程序。

将上述 3 个模块集成到一个应用程序中，通过实现 OPC 服务器开发工具包的特定回调函数来完成开发工具包和通信接口的沟通。服务器调用开发工具包的流程如图 3 所示。

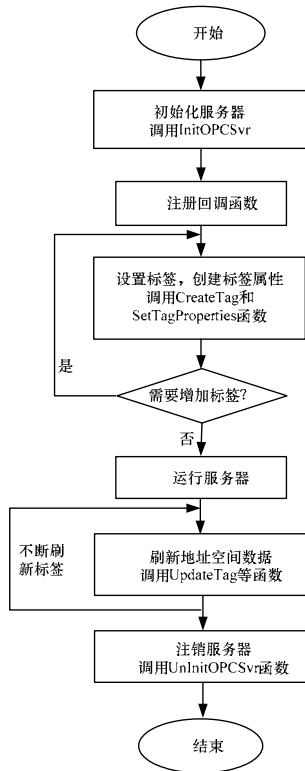


图 3 调用 OPC 开发工具包的流程

调用 OPC 开发工具包的具体步骤如下：

- (1)用户调用 InitOPCSvr 进行初始化服务器，注册回调函

数。(2)用户通过调用 CreateTag 向 OPC 运行库按实际需要添加多个标签并通过 SetTagProperties 函数设置标签属性。(3)运行服务器。服务器开始向客户端传送数据，并调用 UpdateTag 函数不断刷新标签。(4)在服务器退出时，调用 UninitOPCSvr 关闭服务器。

4 软件测试

测试主要针对 EPA 现场总线上的设备(有线和无线)，测试过程是 OPC 服务器将收到的 EPA 现场设备采集到的数据传送给 OPC 客户端，并显示数据。测试系统结构如图 4 所示。

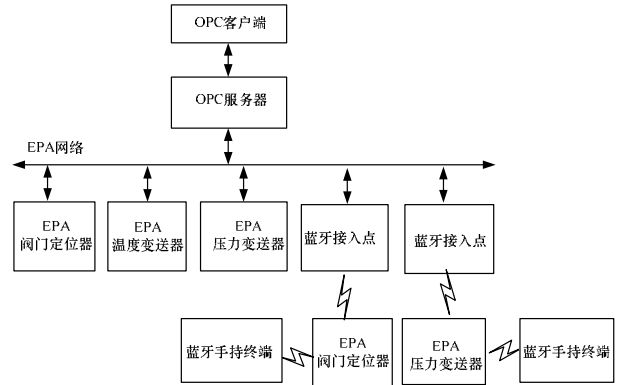


图 4 测试系统结构

运行 OPC 服务器，由服务器界面可知服务器添加了 5 个标签，这 5 个标签代表了和硬件对应的 I/O 点，它们的值是通过 OPC 服务器程序中的驱动程序从硬件里读取的数据，由此得到，标签的时间戳和质量戳。且 OPC 服务器能正确地从现场设备得到数据，进行实时刷新。测试证明了 OPC 开发工具包软件的正确性。

OPC 服务器界面所监视到的 EPA 设备的显示界面如图 5 所示。



图 5 OPC 服务器显示界面

5 结束语

本文介绍了采用 OPC 服务器开发工具包开发的 OPC 服务器的优点。阐述了 OPC 服务器开发工具包的设计过程，并基于该工具包设计了 OPC 服务器，给出 OPC 服务器的设计结构和对工具包进行调用的过程。通过构建一个完整的测试系统验证该 OPC 服务器的正确性。

参考文献

- [1] 马增良, 兰 斌. OPC 数据访问服务器实现机制研究[J]. 计算机工程与应用, 2003, 39(21): 65-67.
- [2] 李 勤, 党选举, 向 荣. 基于 COM 的 OPC 技术的研究和实现[J]. 微计算机信息, 2002, 22(16): 18-22.
- [3] 林 跃. OPC 数据服务器开发工具研究与实现[J]. 自动化仪表, 2001, 22(9): 7-13.
- [4] OPC Foundation. Data Access Custom Interface Standard Specification 2.0[Z]. 2005.

编辑 陆燕菲