

基于 SQL Server 的硬件控制系统设计与实现

刘凌宇, 李晓风, 谭海波, 许金林

(中国科学院合肥物质科学研究院 信息中心, 安徽 合肥 230031)

摘要:针对目前主流商业数据库引擎并不具有工业总线通讯和硬件控制的功能,由此造成了工业现场部署的困难和复杂度问题,文中提出一种基于 SQL Server 进行硬件控制的方法。本方法利用 SQL Server 扩展存储过程的特性,通过在存储过程中封装对 CAN 总线系统中驱动程序接口的调用,来实现 SQL Server 对硬件系统的访问控制,把 SQL Server 从传统的关系数据库应用领域拓展到了数据库管理和硬件控制应用相结合的场合。工程实践表明,本方法为使用单位节约了大量硬件开销,减少了控制总线的设计和部署的困难。

关键词:CAN 总线;SQL Server;扩展存储过程;硬件控制

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2011)10-0047-04

Design and Implementation of Hardware Control System Based on SQL Server

LIU Ling-yu, LI Xiao-feng, TAN Hai-bo, XU Jin-lin

(Information Center, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China)

Abstract: Focuses on the leading commercial database engines lack the abilities of hardware control and industrial bus communication, resulting in difficulties and complexity of database deployment in the industrial fields. Proposed a new method to handle hardware control based on SQL Server. By using the feature of extended-stored-procedure of SQL Server and encapsulation of calling driver interface of CAN-bus system, an example of implementing the CAN-bus communication between the database engine and hardware control units has been given, and the SQL Server has been extended from the traditional relational database applications to the combination of database management and hardware control areas. The project shows that this method saves a lot of hardware costs, reducing system complexity and difficulties of deployment.

Key words: CAN-bus; SQL server; extended-stored-procedure; hardware-control

0 引言

随着自动化要求的不断提高,大量产生的现场和过往数据往往需要一个强劲的数据库引擎来维护,而原有的基于文件等的数据库管理模式已经不能满足要求。并且随着网络技术的发展,原来的单机处理往往被拓展到了 C/S 结构、B/S 结构,从而可以实现更加复杂的控制流程或提高并发度。而在这些领域的应用往往需要数据库管理系统不仅能对数据进行管理,而且还要能根据设定直接控制相应的硬件。而 SQL Server 本身只是一个基于关系型数据库的管理引擎,主要用于对海量数据进行管理维护^[1],并不具备硬件

控制的能力。

所以,文中提出了一种由 CAN (Controller Area Net) 总线构建设备网,通过 SQL Server 的扩展存储过程来实现硬件控制,从而把 SQL Server 的强大的数据管理能力和硬件控制结合起来的新方法。

1 系统设计方案

1.1 传统设计思路

传统的设计方法通过与数据库服务器的交互获得需要的数据,然后由客户端程序来实现硬件的操控,原理如图 1 所示(为简化结构,忽略了前向通道)。这种设计实际上是单机处理和控制的一个扩展,具有上手容易的特点。但是缺点也很明显,每个客户端机器都要部署相应的协议转换硬件,而且还需要部署相应的控制总线和设计较为复杂的控制总线协议。如果硬件成本高或客户端机器数目多、分布地域较广的话,成本将会急剧攀升。而在这种模式下,并不能很好地利用

收稿日期:2011-03-04;修回日期:2011-06-05

基金项目:中科院合肥物质科学研究院院地合作项目(1183010571 F1)

作者简介:刘凌宇(1982-),男,河南伊川人,硕士研究生,研究方向为网络应用开发;李晓风,研究员,博士生导师,研究方向为软件工程、网络管理、计算机自动控制。

SQL Server 自身的锁机制,可能会造成潜在的总线冲突。

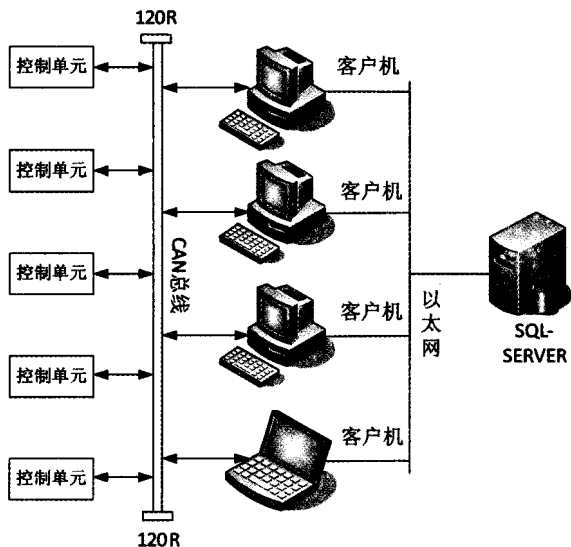


图1 传统设计思路原理图

1.2 改进方案

使用 C/S 或 B/S 结构,将硬件控制功能放在服务器端实现。B/S 结构对客户端的配置要求较低,而且客户端维护量小,程序分发简单,更新维护方便。但 B/S 结构在客户端对大量数据进行实时计算和处理,以及实现图像处理等复杂应用方面,难以满足实时性要求^[2]。因此,在这些应用环境中,还需要考虑采用 C/S 结构。

基于以上分析,提出一种新的设计思路,即利用 SQL Server 直接控制硬件。客户端不再和控制总线相连接,而是通过和 SQL Server 通讯来请求 SQL Server 对硬件的操作。原理图如图 2 所示。

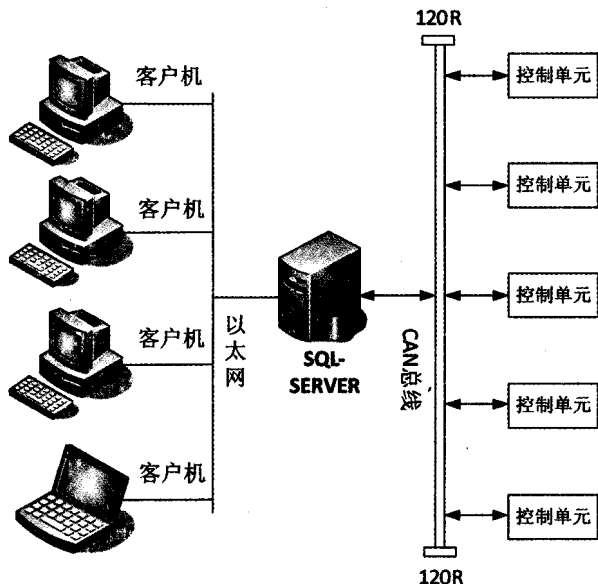


图2 改进的设计方案

从这两种模式的对比可以看出,新的设计可以免

去客户端相关硬件和总线铺设的开销。并且由于往往只有一个主控设备,一般不存在总线冲突等问题,所以可以简化总线的通讯协议。

为实现 SQL Server 对硬件的控制,需要利用 SQL Server 的扩展存储过程。扩展存储过程允许用户在 SQL Server 存储过程中访问用户自己创建的 DLL (Dynamic Link Libraries)^[3]。可以把对硬件的访问封装在自己设计的 DLL 中,然后在相应的存储过程中封装对该 DLL 的调用。客户端通过和 SQL Server 通讯,将有关参数传递给相应的存储过程,就可以由 SQL Server 完成对硬件的操作了。

2 CAN 总线系统设计

CAN 即控制器局域网,属于现场总线的范畴,是一种支持实时控制和分布式控制的异步串行通讯网络,以其优良的特性和突出的可靠性,被公认为最有前途的现场总线之一,应用范围遍及从高速网络到低成本的多线路网络^[4]。

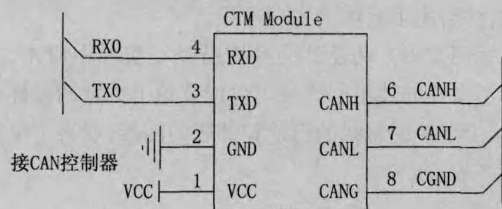
本系统为一个仓库的自动分拣及监控系统,部分客户机用于与货主交互,当货主通过相关验证后,服务器通过与控制单元的通讯,将货物准确的输送到货主身边。同时,部分客户机负责监控前向通道,包括温度、湿度等,并定期的将数据输入服务器,以方便在所有机器上随时检阅仓库状态。如图 2 所示,本系统中,货柜的控制单元通过 CAN 总线互连,上位机为 SQL Server 服务器,通过 PCI-CAN 接口卡与 CAN 总线通讯。由于 CAN 网络上任何一个节点均可作为主节点主动地与其他节点交换数据,因此,当任一控制节点出现异常时,可以及时反馈给服务器,以便管理人员及时处置。

2.1 CAN 节点设计

在以往的控制单元设计中,一般采用两个高速光耦(如 N137),实现电气上的隔离,一个电源隔离模块(+5V 转+5V),实现电源上的隔离,另外还需要计算电阻值的大小才能搭建出合理的收发器隔离电路。需要注意在进行光电隔离时还要保证器件两侧所用电源的完全隔离,否则,光电隔离将失去意义^[5]。由于这种方式存在着体积偏大、成本偏高、采购不便等缺点,因此本系统采用了隔离 CAN 收发器模块,如图 3 所示。

CTM 系列模块集成电源隔离、电气隔离、CAN 收发器、CAN 总线保护于一体的隔离 CAN 收发器模块。该模块 TXD、RXD 引脚兼容+3.3V 及+5V 的 CAN 控制器,不需要外接其他元器件,直接将+3.3V 或+5V 的 CAN 控制器发送、接收引脚与 CTM 模块的发送、接收引脚相连接。采用隔离 CAN 收发器,可以很好地实现 CAN 总线上各节点电气、电源之间完全隔离和独

立,提高节点的稳定性和安全性^[6]。



集电源模块、高速隔离、CAN收发器、ESD保护于一体

图 3 隔离 CAN 收发器模块

2.2 CAN 总线与 PC 通信

本系统使用的 CAN 接口卡是致远电子的 PCI-9820I 工业级 2 通道 CAN 接口卡,每个 CAN 通道均集成独立的隔离保护电路。CAN 控制器为 SJA1000T,兼容 CAN2.0A/B 规范。

通常应用 VC++ 实现串行通讯有如下几种方法:一是利用 Windows API 通讯函数;二是利用 VC 的标准通讯函数 `inp`、`inpw`、`inpd`、`outp`、`outpw`、`outpd` 等直接对串口进行操纵;三是使用 Microsoft Visual C++ 的通讯控件 (MSComm);四是利用第三方编写的通讯类。本系统采用的是该接口卡提供的 Virtual CAN Interface (VCI) 驱动程序接口,包括三个库函数文件: `ControlCAN.h`、`ControlCAN.lib`、`ControlCAN.dll`^[7]。在上位机中,采用 VC++ 来实现对硬件的访问控制(封装成 DLL),通过调用以上库文件,可进行 CAN 总线的相关操作。然后通过 SQL Server 的扩展存储过程调用对应的 DLL 文件来访问 CAN 总线,即实现 SQL Server 对控制单元的访问控制。

调用过程如图 4 所示^[8]。

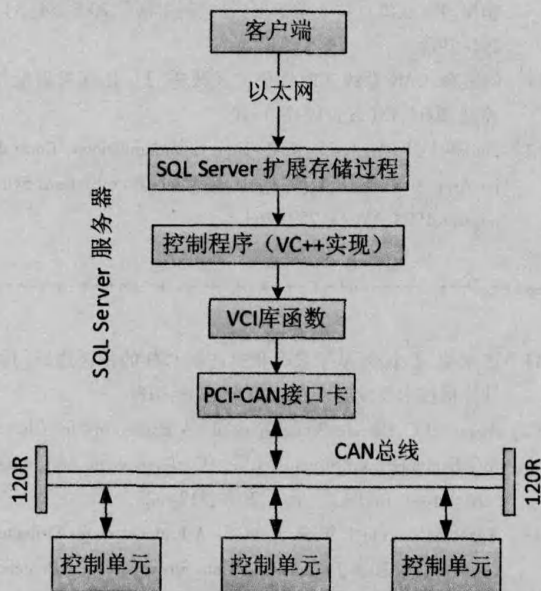


图 4 SQL Server 硬件控制调用过程

3 系统实现

3.1 VC 中实现硬件访问控制类

在 VC 中我们可以创建相应的 ACTIVEX DLL,然后在类中创建访问 CAN 总线的全局函数,作为该类的方法。例如,初始化及启动 CAN 总线设备,通过 CAN 总线发送数据和读取数据、设备参数设置等函数^[9]。然后,编译生成相应的 DLL,将该 DLL 拷入对应的 SQL Server 的 Bin 目录,并完成注册。这样就可以实现在 SQL Server 中调用的 DLL。下面以启动 CAN 设备函数为例进行说明。

```
public int StartCan(int DeviceInd, int CanInd)
{
    int nDeviceType = 16; // PCI9820I
    int nDeviceInd = DeviceInd; // 设备索引号
    int nCANInd = CanInd; // 第几路 CAN
    int nReserved = 9600; // Baudrate
    VCI_INIT_CONFIG vic;
    DWORD dwRel;

    dwRel = VCI_OpenDevice(nDeviceType, nDeviceInd, nReserved); //设备打开
    if (dwRel != STATUS_OK) {
        MessageBox(_T("打开设备失败!"), _T("警告"), MB_OK|MB_ICONQUESTION);
        return 0;
    }

    dwRel = VCI_InitCAN(nDeviceType, nDeviceInd, nCANInd, &vic); //初始化设备
    if (dwRel == STATUS_ERR) {
        VCI_CloseDevice(nDeviceType, nDeviceInd);
        MessageBox(_T("初始化设备失败!"), _T("警告"), MB_OK|MB_ICONQUESTION);
        return 0;
    }

    dwRel = VCI_StartCAN(nDeviceType, nDeviceInd, nCANInd); //启动设备
    if (dwRel == STATUS_ERR) {
        VCI_CloseDevice(nDeviceType, nDeviceInd);
        MessageBox(_T("启动设备失败!"), _T("警告"), MB_OK|MB_ICONQUESTION);
        return 0;
    }

    return 1;
}
```

3.2 SQL Server 扩展存储过程设计

SQL Server 扩展存储过程和一般存储过程相比^[10],具有相同的执行过程和调用方法。不同点在于一般存储过程是一组用存储过程名标识的 SQL 语句集,其功能受到 SQL 语言的限制。而扩展存储过程是 SQL Server 实例可以动态加载和运行的 DLL,使用

SQL Server 扩展存储过程 API 编写的,可直接在 SQL Server 实例的地址空间中运行。它可以存取数据库中的数据(通过 ODBC,OL EDB,DAO,ADO 等),可以实现复杂的分析功能(如图像分析、几何运算等),也可以实现硬件通讯等等。

在本例中使用 SQL Server 调用上面生成的 DLL,首先使用系统存储过程 sp_OACreate 来初始化一个该类的实例,然后利用 sp_OAMethod 来调用实例的相应方法。

给出代码如下:

```
declare @ hr int,@ src varchar(255),@ desc varchar(255)
declare @ obj int,@ StartResult int
exec @ hr=sp_OACreate 'ClassCanbus.CanOperate', @ obj out
if @ err<>0 goto Err --如果创建失败,则进行错误处理
exec @ hr=sp_OAMethod @ obj,'StartCan',@ StartResult out,
0,0
if @ err<>0 goto Err --如果调用错误,则进行错误处理
exec sp_OADestroy @ obj
return
Err:
exec sp_OAGetErrorInfo 0,@ src out,@ desc out
select cast(@ err as varbinary(4)) as 错误号,@ src as 错误源,@ desc as 错误描述
```

3.3 对于上述实现的补充

实际中,为了防止客户端对硬件操控的并发请求造成总线冲突,我们往往还需要采用数据中某个特定字段作为标志位(semaphore),采用锁机制和 ID 号排序的方式来保证对总线的序列访问。并且还要考虑到控制过程可能发生的故障,做好容错处理和收集反馈错误信息。此外,针对网络扩展和网络负载变化引起的信息传输时延,可以考虑使用时分复用技术来改进 CAN 总线的实时性^[11,12]。

4 结束语

文中针对 C/S 结构的现场总线系统中,进行硬件控制系统拓展时面临的复杂度和成本问题,提出了一

种通过 SQL Server 的扩展存储过程来实现硬件控制的方法,并给出了具体的实现过程。

采用文中方法设计的系统已经在实际中应用,并发挥了良好的效果。为使用单位节约了大量的硬件开销,减少了控制总线的设计和部署的困难,取得了良好的经济效益。

参考文献:

- [1] Microsoft Corporation. Microsoft SQL SERVER BOOK ONLINE[DB/OL]. [http://msdn.microsoft.com/en-us/library/ms130214\(v=sql.90\).aspx](http://msdn.microsoft.com/en-us/library/ms130214(v=sql.90).aspx).
- [2] 刘 慧. 基于 C/S 与 B/S 的煤矿安全监察信息管理系统研究[D]. 西安:西安科技大学,2005.
- [3] Sunderic D. Microsoft SQL Server 2005 Stored Procedure Programming in T-SQL & .NET[M]. [s. l.]: McGraw-Hill Osborne Media,2006.
- [4] 邬宽明. CAN 总线原理和应用系统设计[M]. 北京:北京航空航天大学出版社,1996.
- [5] 谢子青. 光电隔离抗干扰技术及应用[J]. 现代电子技术,2003(13):33-37.
- [6] 致远电子. CAN 节点设计[DB/OL]. <http://www.zlgmcu.com/download/downs.asp?ID=2022>.
- [7] 致远电子. CAN-BUS 通用测试软件及接口函数库使用手册[DB/OL]. <http://embedcontrol.com/download/downs.asp?ID=776>.
- [8] 江晋剑. 基于 AT89C55 和 CAN 总线的智能节点设计[J]. 计算机技术与发展,2008,18(4):214-216.
- [9] 姜兴刚. 基于 PCI 总线的 CAN 通信卡的设计与实现[J]. 测控技术,2004,23(7):42-43.
- [10] 万 波. SQL SERVER 扩展存储过程实现机制及应用方法初探[J]. 武汉科技大学学报(自然科学版),2001,24(3):294-297.
- [11] 韩进军. CAN 总线实时性研究及改进[J]. 计算机测量与控制,2005,13(11):1247-1249.
- [12] Tindell K W. Analysing Real-Time Communications: Controller Area Network (CAN)[J]. IEEE Real-Time Systems Symposium,1994,23(7):259-263.

(上接第 46 页)

- ring by predator prey particle swarm optimization[C]// IEEE Congress on Evolutionary Computation. [s. l.]: [s. n.], 2007:3232-3238.
- [9] Jarbou I B, Cheikh M, Siarry P, et al. Combinatorial particle swarm optimization (CPSO) for partitional clustering problem [J]. Applied Mathematics and Computation,2007,192(2):337-345.
- [10] 张爱华,江中勤,张 华. 基于粒子群优化算法的分形图像压缩编码[J]. 计算机技术与发展,2010,20(2):21-24.

- [11] 张家柏,王小玲. 基于聚类和二进制 PSO 的特征选择[J]. 计算机技术与发展,2010,20(6):905-918.
- [12] Aggarwal C, Han J, Wang J, et al. A Framework for Clustering Evolving Data Streams[C]// Conference on Very Large Data Bases. Berlin:[s. n.],2003:312-323.
- [13] Aggarwal C, Han J, Wang J, et al. A Framework for Projected Clustering of High Dimensional Data Streams[C]// Proceeding of the 30th very large data bases conference. Toronto:[s. n.],2004:852-863.