

# 基于协议栈的现场总线协议转换通用化设计

蒋浩天, 雷 航

(电子科技大学 计算机科学与工程学院, 四川 成都 610054)

**摘 要:**为实现不同现场总线之间的通信和提供统一的设计模式,达到屏蔽现场总线异构性的目的,对多种现场总线集成环境下不同总线之间的通信进行了研究,建立协议栈模型实现了协议转换网关的通用化设计。与传统的协议转换方式相比,不再局限于一对一的转换,具有与对象位置、通信协议的无关性,提高了系统的扩展性和对不同总线的兼容性。

**关键词:**现场总线;协议转换;协议栈

**中图分类号:**TP302

**文献标识码:**A

**文章编号:**1673-629X(2007)11-0015-04

## Universal Design of Multi-Fieldbus Protocol Conversion Based on Protocol Stack

JIANG Hao-tian, LEI Hang

(Department of Computer Science & Engineering, University of Electronic Science  
and Technology of China, Chengdu 610054, China)

**Abstract:** In order to implement the communication of various kinds of fieldbuses and supply the uniform design model so that the heterogeneity of the fieldbuses can be shield, the communication under fieldbuses integrative network is researched and the protocol stack model is built to implement unification design of interconnect gateway. Compared with old conversion method, it's not limited to one-to-one conversion, and independence of the object location and communication protocols, which improves system's augment ability and compatibility for various kinds of fieldbuses.

**Key words:** fieldbus; protocol conversion; protocol stack

### 0 引 言

随着控制、计算机、通信、网络等技术及智能传感技术的发展,信息交换沟通的领域正在覆盖从工厂的现场设备到控制、管理的各个层次。现场总线就是顺应了这个形式发展起来的新技术,它应用在生产现场,在微机化测量控制设备之间实现双向串行多节点数字通信的系统,也被称为开放式、数字化、多点通信的底层控制网络<sup>[1]</sup>。在工业控制的现场,不同时期和不同的环境下可能会安装着支持不同标准的、具备各种属性的现场总线设备,据不完全统计,目前国际上有 200 多种总线,占据市场主流的有十几种,其中市场份额较大的有 CAN、Profibus、FF(基金会现场总线)、Lonworks、WorldFIP 和 P-NET 总线。各类现场总线的应用范围如图 1 所示<sup>[2]</sup>,由图示可知,没有哪种现场总

线能够完全适用于所有的应用范围,但是不同总线之间又存在交集,这造成了一个应用环境中存在多种总线,如 CAN、Profibus、Lonworks 在楼宇自动化场合都有很好的应用。随着各种总线技术的不断发展,在不同应用领域中,已经越来越多地要求异构现场总线网络和服务上的集中,并向用户提供统一的访问接口,使得能够方便地控制和访问异构总线网络。目前,在现场总线协议转换方面的研究成果,主要是利用 OPC 技术实现过程控制级转换,以及利用嵌入式网关实现总线一对一的协议转换。采用 OPC 接口技术可以实现不同总线之间的转换,所有工作由上位机中软件完成,实现方式简便,但缺点也较明显:由于依赖上位机,总线系统中节点间的通信较慢,实时性能不够;对微软操作系统有很强的依赖性,使得它在嵌入式系统中的实现有一定难度;由于需要配置 OPC 标记,基于 OPC 的控制系统需要更多的开发时间。

目前使用最为普遍的是网关式转换,将协议转换的工作在总线中完成<sup>[3]</sup>,可实现差异较大的总线协议之间的转换<sup>[4]</sup>。目前研究和实现的嵌入式网关通常是

收稿日期:2007-01-19

**作者简介:**蒋浩天(1981-),男,四川乐至人,硕士研究生,主要研究方向为现场总线网络,嵌入式应用;雷 航,博士,副教授,主要研究领域为嵌入式系统实时性和可靠性。

一对一的协议转换,有一定局限性,在实现方式和软件设计上没有建立一个统一模型,考虑新的总线势必对软硬件模块进行重新设计,不便于总线扩展和网关功能扩充。为实现网关的统一化设计和可扩展性,文中提出了一种基于类协议栈的嵌入式网关的软件设计模式。

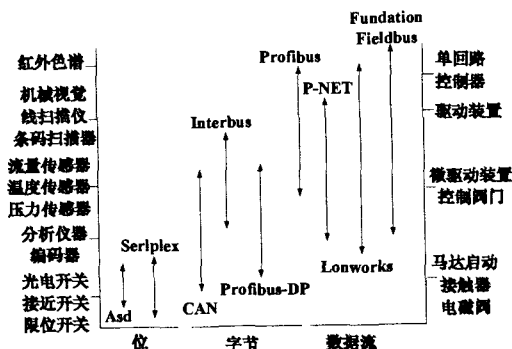


图 1 各种现场总线的应用范围

## 1 现场总线协议转换技术分析

### 1.1 OPC 技术和协议转换

OPC 是控制系统现场设备级与过程管理级进行信息交互的开放接口标准和技术规范,由 OPC Foundation 机构维护和管理,目前已经推出 OPC 数据和访问、报警与事件和历史数据规范。它采用客户/服务器模式,以 OLE/COM 机制作为应用程序级的通行标准,将开发访问接口的任务放在硬件生产厂家或第三方,以服务器的形式提供给客户,并规定了一系列的接口标准,客户负责创建服务器对象和访问服务器支持的接口。它主要解决服务器和客户端的实时数据存取问题,从现场设备读取数据,把来自不同硬件供应商的不同类型数据转换为统一的 OPC 数据格式,以 OPC 接口的方式传送给客户应用程序,从而实现系统信息的集成。

OPC 服务器是作为现场结构的中间层存在的。由于采用的客户/服务器模型,因此一个客户应用程序可以和多个硬件厂商提供的 OPC 服务器相连,同时一个 OPC 服务器可以和多个客户应用程序相连,形成多对多的关系,达到在同一系统中集成不同现场总线类型的目的,系统集成如图 2 所示。

OPC 技术一定程度上实现了多总线互操作性,但也有缺点:由于 OPC 技术以微软的 COM 技术为基础,对上位机和操作系统的依赖性较强,它在具体的嵌入式应用场合,总线节点之间通信的速率不理想,同时它需要厂商的技术支持。

### 1.2 传统嵌入式一对一网关

网关是在不同通信协议、数据格式结构、语言或体

系结构的两个系统间充当转换器角色的计算机系统或其他设备。网关要求具有物理接口功能、通信协议功能和操作信息功能,能够实现不同总线系统之间的信息连接和管理,主要完成帧格式转换功能和发送、接收总线信号的功能。按照 ISO/OSI 参考模型,由于不同的现场总线的物理层、数据链路层定义了不同的传输媒介和数据转发机制,要实现异类总线协议的转换,网关应当工作在协议的最高层应用层,通过对不同协议的数据报文封装和解包来实现。网关需要有对连接的每种总线相适应的物理接口(各种总线控制器或者其他设备),同时需要一个控制器和程序来实现协议转换。一个一对一转换的协议网关结构例子如图 3 所示。

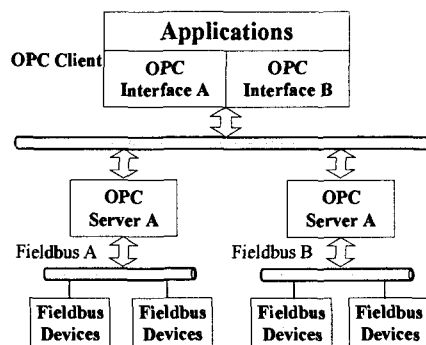


图 2 利用 OPC 实现总线集成的模型

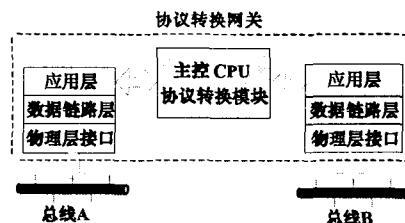


图 3 协议转换网关结构模型

对 A 总线和 B 总线来说,协议转换网关相当于总线系统中的一个设备。在一对一的协议转换网关模型中,协议转换模块功能至关重要,它是各通信程序之间互操作的桥梁,负责对不同总线之间的数据的解包和协议转换的实现。目前,协议转换模块的设计仍然局限于既定的两种总线之间的转换。随着总线系统的升级更新和新的总线设备的加入,势必造成多总线的环境。如:在 CAN 总线、RS-232 总线环境中,添加 Profibus 总线,原有的协议转换模块不再适用,需要重新开发 CAN 对 RS-232, CAN 对 Profibus, RS-232 对 Profibus 的协议转换模块,考虑有  $n$  种总线的环境,如果按照一对一的转换方式,需要开发  $n * (n - 1) / 2$  个模块,每种总线的加入都需要开发对应另外  $n - 1$  种总线的转换,必然造成软件的重复设计和开发效率的降低。如果网关的软件设计能够提供统一、通用的设计

方式,每种总线的加入只针对适应通用化而进行设计,可以大大地降低开发时间;同时可以提供开放兼容的接口,方便用户应用程序的设计。

2 基于协议栈的协议转换的设计

现场总线协议基于精简的 ISO/OSI 7 层参考模型,一般包括物理层、数据链路层和应用层(用户层),省去了第 3~6 层以确保高效的数据传输。如 CAN 总线分为数据链路层(包括逻辑链路控制子层和媒体访问控制子层)和物理层,profibus 使用物理层、数据链路层和用户层。针对多总线综合网络的以上特点,协议转换的结构可以采用类协议栈的方式实现:下层提供一个软件接口,向上屏蔽了具体的实现细节,保证了各层的独立性,有利于模块化设计,减少开发时间和难度。协议栈结构如图 4 所示,在这个协议栈中,为了实现协议转换的统一性和扩展性,在现场总线原有的物理层、链路层基本结构之上增加了一个协议层,为应用程序和信息通信提供统一接口(API),达到屏蔽总线差异的作用。同时,为了实现对网关的实时控制和调度,最上层是应用控制层。对于上层用户来说,协议栈提供了一个访问总线的接口,并屏蔽了下层总线之间的差异。协议的转换就是各种总线的数据在网关中逐层上行,然后又逐层下行到另一总线的过程。

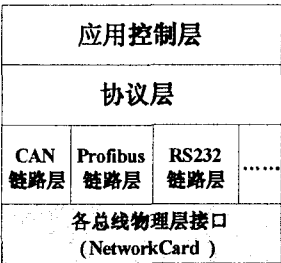


图 4 网关协议栈结构

要屏蔽总线的差异,提供统一的协议转换接口的协议层和应用控制层的实现是设计的关键也是难点,需要解决以下几个主要问题:总线节点地址的一致性;数据交互信息的一致性和应用接口的通用化。下面就这 3 个问题进行分析并给出了解决的办法。

2.1 统一的节点地址划分

在总线扩展或者删减的过程中,系统的节点和数据交换路径都可能发生改变,因此,统一的节点访问方式可以解决网络扩展中通信路径变化的问题。

网关对现场设备采用统一编址,由物理层向上屏蔽具体总线的差异。在现场总线网络中的节点都有统一的逻辑地址和物理地址,每个总线上的节点都可以使用统一的逻辑地址进行通信。网关为每种总线分配固定的逻辑地址,逻辑地址为 16-bit,其首 4-bit 用

来标识网络类型,后 12bit 用来标识节点。而物理地址和典型的各总线地址相似,以 CAN 总线为例,CAN 的目的地址将与之标识符 ID 相对应。系统维护一个逻辑地址-物理地址的转换表 AddrTable,在协议转换中,节点之间交换数据使用对方的逻辑地址。当协议封装的时候需要使用到与总线类型相关的物理地址,这时由协议转换模块查找 AddrTable 获得。表 1 为一个 AddrTable 的实例。

表 1 AddrTable 的例表

逻辑地址(CAN 总线)	物理地址(CAN 总线)
0x1019	0x19
0x1020	0x20
0x1016	0x16
逻辑地址(Profibus 总线)	物理地址(Profibus 总线)
0x2016	0x16
0x2017	0x17
0x2018	0x18

对于 CAN 总线上的节点来说,要读取逻辑地址 2016H 位置的数据,首先网关分析协议数据中变量地址部分,确认要访问的节点是 Profibus 总线地址节点的,于是查找 AddrTable 表找到目的物理地址,利用该地址发出包含数据请求的帧,网关接受 Profibus 总线传回的数据,查找到 CAN 节点对应的标识符(物理地址),以 CAN 总线的标准进行协议数据单元封装并发送。对于 CAN 总线上请求数据的节点,它虽然不知道目标节点的具体位置,但像在同一总线操作一样,而不用关心数据来自哪条总线的哪个节点和传输过程。

2.2 数据交互信息统一化

在一对一转换中,通过对两种不同协议帧的分析,可以直接在两种总线协议帧之间建立映射关系。多种不同总线存在的情况下,不能局限于一对一的转换,否则每加入一个总线形式都需要针对其他的所有的总线类型进行映射关系的添加,可扩展性和兼容性较差,所以首先必须定义一种统一的数据类格式,新的类型只需要针对这个统一类进行映射。这个数据类格式被当作协议层的帧。通常的数据项包含协议类型、网络中地址、设备相关功能和协议数据单元等。协议层统一帧格式如图 5 所示。

SrcAddr	DstAddr	prio	length	timer	timer	type	data
---------	---------	------	--------	-------	-------	------	------

图 5 协议层统一帧格式

SrcAddr 和 DstAddr 分别代表了源、目的地址,prio 为信息优先级,timer 是为实时调度考虑加入的表项(包含开始时间和 deadline 等),type 为事先定义好的操作类型,写数据或者数据请求。

统一帧数据结构示例如下:

typedef struct UNIFYDATA

```

{
    unsigned short SrcAddr; // 发送节点逻辑地址
    unsigned short DstAddr; // 目的节点逻辑地址
    unsigned int prio; // 帧信息优先级
    unsigned int length; // 缓冲区中数据长度

    timer ctimer;

    unsigned char type;
    unsigned short data[8];
}

typedef struct timer
{
    long starttime;
    long deadline;
}

```

timer 数据结构用于应用控制层对协议层的统一帧数据的实时性进行控制,为了保证数据在 deadline 之前发送到对应的总线上,应用控制层应当设立一个监听程序对所有的数据请求、数据交互进行监听,利用优先级和时间约束进行帧排序,把排序的结果放入一个帧链表当中,优先级高的帧插入链表表头抢先获得处理。

### 2.3 统一的应用接口

对于协议层的统一帧,需要各总线有对应的读写控制函数接口,向上层用户和应用提供统一的 API。在协议转换程序中,用轮询各总线方式或者利用中断来监听各总线的的数据请求。每种总线至少需要 3 个接口函数:xxxRead, xxxWrite, xxxOpen, xxx 代表具体总线类型。为了更好的完成用户任务和协议转换任务,提高系统实时性能,本文采用 VxWorks 操作系统驱动方式来实现用户访问统一接口,向用户层屏蔽下层具体总线接口函数实现细节的差异。VxWorks 是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统(RTOS),是目前世界上用户量最大的实时操作系统<sup>[5]</sup>。

VxWorks 通过 I/O 系统访问设备驱动,它的设备驱动采用了文件描述符(FD\_TABLE)、设备列表(DEVICE LIST)和驱动表(DRV\_TABLE)的机制。我们需要为新设备设计设备描述符,在设备链表中插入新设备,为新设备注册驱动程序,并为每一种总线类型定义的驱动接口中包含 VxWorks 的 I/O 操作函数,如 can 总线需要接口有如下的函数:

Canbus interface:

```

{
    boolean canDevCreat ( );
    // can 总线设备初始化和创建
    boolean canOpen ( ); // 设备打开
    boolean canWrite (UNIFYDATA data);

```

```

// 向 can 总线发送统一帧格式报文并转化为 can 帧
UNIFYDATA canRead (UNIFYDATA data );
// 从 CAN 总线接收报文并转化为统一帧
canIoclrCan ( ); // 总线 I/O 控制
.....
}

```

这些函数的具体实现要根据 VxWorks 运行的平台和微控制器的不同作修改,他们与不同的物理设备打交道,将成为 VxWorks 系统 BSP 的一部分。有了以上的接口函数,用户可以不用关心对总线设备的操作,在协议转化时,直接利用 xxxRead 函数从总线上读取数据信息,经过应用层主任务对统一帧的分析,取出地址信息,查表并调用目的总线的 xxxWrite 函数把信息发送出去。协议数据转换任务的流程如图 6 所示。

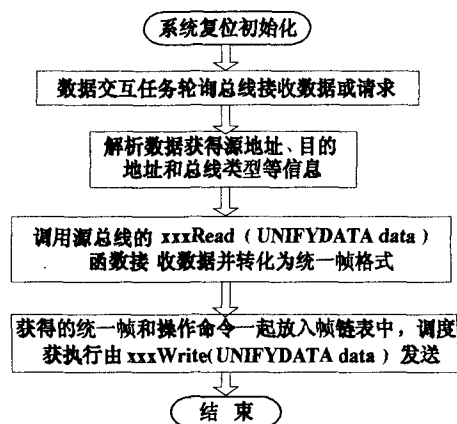


图 6 协议数据转换流程

这样设计针对多总线的通用化表现在:

- 1) 由于是在应用层进行协议的转化,用户对新加入的总线类型可以直接开发该总线的驱动,而不用考虑其他总线的差异;
- 2) 统一帧格式为接口的统一化提供了便利,当上位机需要对现场信息进行监控时,可以把它做为“用户数据”直接嵌入到 TCP/IP 数据帧中,然后通过以太网进行传送;
- 3) 当总线更新发生节点的增减而造成通信路径改变的时候,只需要对地址表 AddrTable 进行修改。

### 3 结束语

多总线综合网络的集成是自动化领域研究的热点,实现不同总线之间的通信,屏蔽底层总线的差异有十分重大的意义。文中利用嵌入式网关协议栈的方式为多总线协议数据转换提供了一个通用化和可扩展性较好的方案,可以避免重复设计。在实验室条件下实现的多总线网关运用了以上的设计技术。由于总线硬件环境的不同和应用层协议的差异,在系统设计的最初必须考虑尽可能多的总线,这为网关的设计增加了

(下转第 22 页)

## 2.4 特征重定位机制

在协同设计的过程中,采用的是分布式结构,设计人员分布在不同的客户端,而且各自都是完成一些局部的特征造型,然后再提交给服务器进行进一步的集成操作。按照特征重排的原理设计出这个系统,保证了客户端在提交特征量时无需按照原始顺序提交,从而大大提高了系统的灵活性。

对于每一次特征操作,客户端的交互过程如下:一个设计者(即客户端)首先设计出特征模型,然后选取合适的位置和方向约束(即定位特征和方向)。当这些完成后,设计者向服务器提交特征信息,并获得显示结果。该显示结果是由服务器进行相应的布尔操作后传送回客户端的。当有多个客户端时,由于任务被分解,在服务器端接收到的特征顺序就会被打乱。如果服务器还是按照原始操作执行,将会得到一个错误的结果。所以此时的布尔操作需要考虑到特征重排的顺序问题。

当一个特征到达时,首先对其进行特征重排。协同管理器利用每个特征量的唯一编号结合基于特征的重排思想实现设计结果的正确性。当服务器正在处理一个客户端的提交时,它首先提取出此提交特征  $F_i$  的编号,并将其与已经处理的特征编号进行比较。只有那些本来应该在其后处理而现在提前处理了的特征才有可能影响对此特征的处理。因此需要对此特征进行重排,以防止它与这些“反序”(相对于此特征前后顺序发生变化)的特征出现不必要的操作。具体的重排过程是:对所有编号大于它的特征  $F_j$ ,  $F_i$  的布尔操作改为:

$$\otimes_i V'_i = \otimes_i (V_i - \sum \varphi(\otimes_i, \otimes_j) V_j) \quad (5)$$

$$\text{这里: } \varphi(a, b) = \begin{cases} 1 & a \neq b \\ 0 & a = b \end{cases}$$

就是用这些  $F_j$  对  $F_i$  进行重新修剪然后再与全局特征量进行布尔操作。若不存在这样的  $F_j$ , 则不进行修剪,即  $\otimes_i V'_i = \otimes_i V_i$ 。用  $M_n$  表示对  $n+1$  个实体模型应用  $n$  次布尔操作后得到的最终模型:

$$M_n = \prod_{i=0}^n \otimes_i V'_i \quad (6)$$

经过特征重排之后,特征造型的结果仍然是正确的。采用特征重排的思想可以很大程度地提高协同设计的效率。在本系统模型中,将特征重排与令牌机制结合起来,一方面可以保证整个协同的过程井然有序,另一方面使得服务器无需等待特征的按序到来,可以大大提高协同设计的效率。

## 3 结束语

随着计算机网络技术的快速发展和特征建模技术研究的深入,计算机支持的协同特征建模将是 CAD 技术发展的一个重要方向,也是协同设计理论研究与发展的必然趋势。因为当前多数基于特征的协同设计造型系统不能异步处理多个客户端提交上来的特征造型。文中主要讨论了客户端不必按照原始的特征造型顺序来提交,先完成造型任务的可以先提交,而造型的结果却能保持与原始建模结果完全一样的设计,所提出的这种重排特征顺序的模型思想有助于增加协同设计系统的灵活性,在协同设计中具有重要意义。但是,文中所提出的模型和设计思路还处于理论阶段,还有待进一步实现。

### 参考文献:

- [1] 尚建忠,孟凡磊,唐力,等. 基于三维 CAD 平台的实时协同设计方法[J]. 机械设计, 2005, 22(8): 43-45.
- [2] LEE SANG HUN. Feature-Based Multiresolution Modeling of Solids[J]. ACM Transactions on Graphics, 2005, 24(4): 1417-1441.
- [3] 周自强,李晓光,沈连娟. 基于特征语言的远程协同实体建模研究[J]. 现代制造工程, 2003(2): 24-26.
- [4] 李伟刚,莫蓉,杨海成,等. 基于 Web 的零件特征共享技术研究[J]. 计算机辅助设计与图形学学报, 2002, 14(10): 1-5.
- [5] 汪惠芬,张友良,曹健. 基于特征的协同设计[J]. 计算机辅助设计与图形学学报, 2001, 13(4): 367-372.

(上接第 18 页)

难度。由于协议数据转化主要是软件实现,在软件的实时性能方面,如何定义各总线帧的实时约束和全局优先级,实现优化调度是下一阶段开发过程中进一步研究的重点。

### 参考文献:

- [1] 夏德海. 现场总线现状及其应用[J]. 中国仪器仪表, 1998(3): 2-3.
- [2] 郭宽明. 现场总线应用选编[M]. 北京: 北京航空航天大学出版社, 2004.

学出版社, 2004.

- [3] 丁磊,费敏锐,陈维刚. 现场总线协议转换机理及实现[J]. 自动化仪表, 2005(7): 1-3.
- [4] Labashov M, Pratl G, Sauter T. Applicability of Internet protocols for fieldbus access[C]//Proc. 4th IEEE International Workshop on Factory Communication Systems. Varsalas, Sweden: [s. n.], 2002: 28-30.
- [5] 张万升,刑涛,刘尉悦. 基于 VxWorks 的多 DSP 系统的多任务程序设计[J]. 单片机与嵌入式系统应用, 2001(6): 9-11.