

一种串行接口设备的 Internet 接入服务器^{*)}

王小英 赵海 阮冠春 张文波

(东北大学信息科学与工程学院 420#信箱 沈阳 110004)

摘要 随着设备网络化及普适计算模式的形成,许多带有传统串行接口的设备产生了 Internet 接入的需求。早期的解决方案要求 Internet 接入服务器能够识别设备的数据通信协议,并依据通信协议的内容来确定网络数据报的报文。为了提高接入服务器的通用性,减少接入服务器对设备通信协议的依赖,本文基于嵌入式 Internet 技术,提出了一种通用 Internet 设备接入服务器。它无需对现有设备做任何修改,不依赖于串行设备自身数据帧格式的结构,“透明”地实现了面向流传输方式的串行接口和面向包传输方式的网络接口之间的数据转换。

关键词 数据帧,串行接口,TCP/IP,嵌入式 Internet,设备接入服务器

A Kind of Internet Access Server for Devices with Serial Interface

WANG Xiao-Ying ZHAO Hai RUAN Guan-Chun ZHANG Wen-Bo

(School of Information Science & Engineering, Northeastern University, Shenyang 110004)

Abstract With the development of device networking and ubiquitous computing, many devices with traditional serial interface have a consumingly requirement of accessing Internet. In many early solutions, Internet Access Server must distinguish device communication protocols and group network datagram according to the detailed content of protocols. For the sake of improving generality of access server and reducing dependence of device protocols, this paper presents a kind of general Internet Device Access Server based on Embedded Internet technique. Whatever data frame structure of exist serial devices, DAS transparently implements the transformation between serial interface data featured by stream transmission and network interface data featured by datagram transmission.

Keywords Data frame, Serial interface, TCP/IP, Embedded internet, DAS

1 引言

“所有设备立即联网”是未来科技发展的趋势。在自动化工业领域有成千上万的传感器、执行器、检测器、PLC 等设备,互相连接形成一个控制网络。而最常用来连接这些设备的通信接口就是 RS-232 和 RS-422/485 总线。串行通信具有连接简单、使用灵活方便、数据传递可靠等优点,在数据采集和实时控制系统中得到了广泛应用。但是随着计算机技术和控制技术的不断发展,原有的通信接口在通信速率和通信距离上已经越来越不能满足需求。在这种趋势下,以 TCP/IP 和以太网为代表的成熟度较高的开放式网络技术,正逐渐应用于各个自动化系统,连接并控制所有的设备。然而由于现存串口设备的广泛使用,对现有设备的网络化改造不可能一蹴而就。如何利用简单方法把现有串口设备进行联网,解决串行接口的各种弊端和限制,将具有特别的意义。

早期的解决方案要求 Internet 接入服务器能够识别设备的数据通信协议,并依据通信协议的内容来确定网络数据报的报文。这样,对于不同的设备,接入服务器都必须采用不同的数据处理方法。为了提高接入服务器的通用性,减少接入服务器对设备通信协议的依赖,本文基于嵌入式 Internet 技术,提出了一种通用 Internet 设备接入服务器 DAS(Device Access Server)。它无需对现有设备做任何修改,不依赖于串行设备自身数据帧格式的结构,“透明”地实现了面向流传输方式的串行接口和面向包传输方式的网络接口之间的数据转换。本文第 2 节概要介绍了设备的三种通讯模式,第 3 节是 DAS 的体系结构,第 4 节对 DAS 实现的几个关键技术进行

了分析,第 5 节比较分析了 DAS 的性能,最后是小结。

2 设备通讯模式

通过对现有串行接口设备的通信模式进行研究分析,同时考虑其相应的网络化改造方案,我们根据通信双方的不同属性,定义了三种不同的通讯模式:隧道模式、套接字服务模式和基于驱动程序模式。

2.1 隧道模式

参加串口通信的双方均为设备,无台式计算机系统的参与。用户通过将 DAS 分别连接到收发数据的现场设备,来建立串行的隧道。一般常见于一些简单控制系统中。比如在采用手持编程器对变频器的控制系统中,手持设备与变频器之间就是按照 RS-232 标准进行通信的,而参加通信的双方均为单片机系统。在对这类设备进行网络化改造的时候,必须在通信双方都采用 DAS 服务器,如图 1 所示。其中 a)为现有的通信方式,设备 A 和设备 B 通过一条 RS-232 信道进行连接。在进行网络化处理时,如图 1 中 b)所示,通信的双方均采用了 DAS 服务器作为网络接口,使得通信双方可以进行跨越 Internet 网络的通信。在 DAS 的设计中,不对 RS-232 接口的数据帧格式进行限制。所以可以适用于任何同类线路的通信,接入 Internet 的设备也不用作任何改动。这种工作模式在充分利用现有设备资源的基础上,还能实现跨越网络的通信。这种改进不仅仅是通信方式的改变,随之而来的是通信双方的空间距离的延伸,以及将由此产生的各种测控和服务方式的变革。

^{*)}基金项目:国家高技术研究发展计划项目(2001AA415320)。王小英 博士研究生,研究方向:计算机网络,嵌入式计算机系统;赵海 教授,博士生导师,研究方向:计算机网络,嵌入式计算机系统。

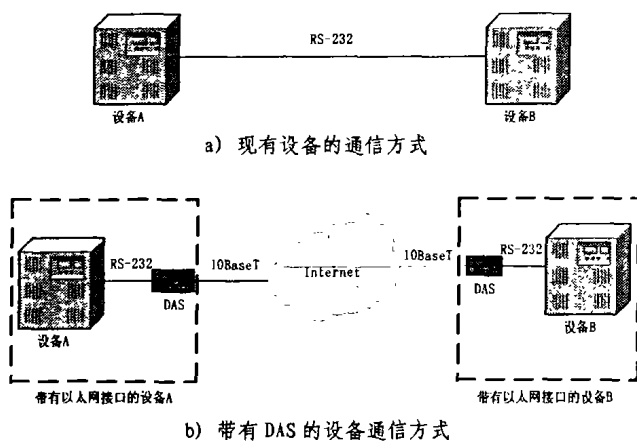


图 1 设备通过 DAS 实现串行通信

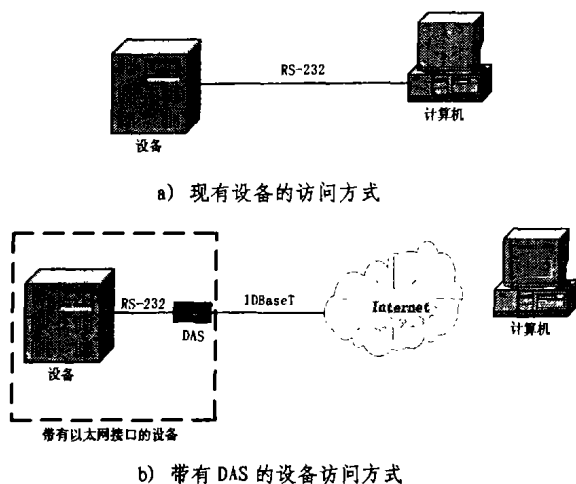


图 2 计算机访问远端串行接口设备

2.2 套接字服务模式

通信双方中有一方为 PC 机系统,且运行着能与串行设备通信的程序。套接字服务模式如图 2 所示,其中 a)表示的是一台计算机通过 RS-232 接口与设备通信,这种通信方式大量用于设备的维护和配置过程。例如对路由器配置时,就需要一台运行有超级终端软件的计算机通过 RS-232 串口与路由器的 console 口进行通信。图 2b)是采用了 DAS 后设备与 PC 机通信。设备的通信速率和信息帧格式仍然不需要改变,由 DAS 完成从 RS-232 接口到以太网接口的转换。而计算机一方的接口形式则由串行接口转换为相应的网络接入形式。整个通信过程不变,参加通信的设备内部程序和结构不需要进行任何修改,但计算机内部现有程序必须做接口部分的修改。

2.3 基于驱动程序模式

DAS 服务器的使用还可以设计为基于驱动程序的控制方法,即通过为设备服务器创建基于驱动程序的控制方法(驱动程序通过在主机上创建一个虚拟本地 COM 或 TTY 端口,在主机和网络串行设备之间建立连接)。将基于服务器/PC 的解决方案提高到新的层次。

三种不同的模式,对于设备端采用的 DAS 服务器来说都是一致的,只是在另一端的数据处理和表现方法不同。本文只研究设备端的 DAS 服务器的实现,而对客户端的数据表现形式不做更多的研究。着重于前两种形式的设计和实现,对于第三种应用形式,重点和难点在于驱动程序的编写工作。必须编写针对不同主机、不同操作系统的驱动程序,并同时进行本地

端口的虚拟映射。

3 DAS 体系结构

3.1 体系结构

DAS 系统为嵌入式系统,硬件上以 ATMEL 公司的 8 位 AVR RISC 芯片 AT90S8515 微控制器为核心,配以 REAL-TEK RTL8019AS 10M 以太网接口芯片,具有 10Base-T 的 RJ-45 接口、14 位 TTL 电平双向 I/O、UART 口、ECP 系统编程口和 DC 5V 电源。软件全部由 AVR 汇编语言编写而成。整个软件系统分为 4 层结构,如图 3 所示。

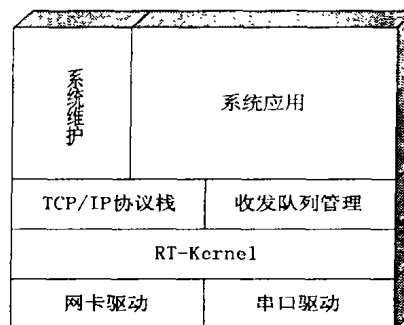


图 3 DAS 系统的软件体系结构

最底层为驱动层,主要由网卡驱动和串口驱动两大部分组成。网卡驱动负责管理 RTL8019AS 网卡芯片的操作,包括端口读写、初始化、网卡缓冲环管理、DMA 操作等几个模块。串口驱动部分负责实现与用户设备的底层通信,主要包括串口收、发中断管理及其相关的服务函数。

中间层主要完成的是针对各自低层数据驱动的协议及队列管理。在实时内核(RT-Kernel)的支持下,针对网络接口需要,实现系统所要求的 TCP/IP 协议栈,包括 ARP、IP、ICMP、UDP 等标准协议以及系统自定义的 SDTP 协议和管理通信协议;针对串行接口,在中间一层需要实现的是收发缓冲队列的管理,主要是针对两个数据缓冲环的操作过程。

高层由两大部分组成,分别是系统应用部分和系统维护部分。系统应用部分为本系统的核心算法实现过程。它一方面要处理来自串口的设备数据,另一方面也处理来自网卡的数据。来自不同端口的数据将按照系统的核心算法处理流程进行处理,以确定最终的处理方式。系统维护部分主要是处理来自网络的维护信息数据报文,为系统正常运行提供参数设置。

3.2 数据报文的处理

DAS 服务器在接入以太网后,有 4 种类型的数据报文:业务数据报文、DAS 系统维护报文、辅助网络协议报文和其它无效报文。它们分别对应三个不同的处理模块:业务数据处理模块、系统维护模块和辅助网络报文处理模块。

3.2.1 业务数据处理模块 一方面对网络到来的业务相关数据报文进行处理,另一方面对串口来的数据流按 SDTP 协议进行封装。通信双方的 IP 地址和端口信息均可以通过维护模块进行配置。

图 4 描述了单方向上的数据传输及处理时序。空闲时 DAS 服务器处于监听状态,同时监听来自串口和网络的数据。ThisDAS 在收到来自串口的第一个数据后开始串行接收计数和计时工作。当这两个条件之一满足时,DAS 系统将按照 SDTP 协议,把串口收到的数据加上一个序号,作为应用层数据以 UDP 的协议方式发送到指定的 PeerDAS 的指定端口,并启动网络超时定时器,以确定是否需要重发以及重发的

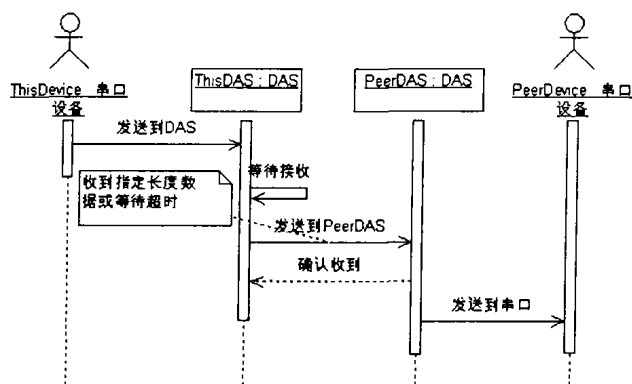


图4 串口数据发送时序图

时间。PeerDAS 收到网络上传来的数据后,将对其序号进行

表1 系统配置参数表

项目	长度(字节)	描述	示例
IP 地址	4	DAS 服务器的本机 IP	192.168.0.12
子网掩码	4	DAS 服务器的本机子网掩码	255.255.255.0
网关	4	DAS 服务器的网关地址	192.168.0.254
本机 MAC 地址	6	DAS 服务器的 MAC 地址	00-06-ee-01-05-09
本机接收端口	2	DAS 服务器接收网络数据的 UDP 监听端口	6001
远程 IP	4	与 DAS 服务器通信的 IP	192.168.1.22
远程接收端口	2	DAS 服务器的数据发送目的端口	6002
串口通信波特率	1	DAS 服务器与串口设备进行通信的波特率	47(9600bps)
串口数据帧长度	2	从串口接收数据的最大帧长	200
串口等待时间	2	指定串口等待的超时时间	50

3.2.3 辅助网络报文处理模块 除了正常的数据报文外,还有一些必需的辅助报文需要处理,包括 ARP 协议报文, ICMP 协议报文。

1. ARP 报文的处理:ARP 报文用于寻找指定的 IP 地址所对应的 MAC 地址。ARP 被划分成两部分,第一部分为发送 ARP 请求报文并处理收到的应答,第二部分是回答其他机器的 ARP 请求。

2. ICMP 报文的处理:使用 ICMP 的回送请求(echo request)报文和回送应答(echo reply)报文来识别网络问题。

除了以上必须处理的报文外,DAS 还会收到来自网络的一些无效报文。比如:寻找其他 IP 的 ARP 广播,MS Windows 的网络主机列表维护广播等。对这些报文,在本设计中对其不进行处理,直接从网卡缓冲区中删除即可。

4 关键技术分析

4.1 数据通信状态机

由于 DAS 面向通用的串口设备,没有固定的数据帧格式,因此从串口上来的所有数据都被看作是以流的方式来进行处理的。但是,网络数据的传输却是以数据报文格式进行处理的,换句话说,网络中的数据是需要分片传输的,这样就产生了如何对数据流进行分片的问题。为了解决这个问题,我们设定了一个通信机制,如图 5 所示。

系统上电后串口处于 Idle 状态。当从串口收到第一个字节后进入 Receive 状态,并启动与接收相关的定时器,用于对接收等待的计时。同时还启动一个记录已收到字节数的计数器。如果从串口收到指定长度的数据或在指定的时间段内没有收到新的数据,系统将对从串口收到的所有数据进行打包

确认,并发送一个确认信息给对方。如果序号正确,系统则把该帧数据以约定的方式发送到 PeerDevice。

以太网和串口设备都工作在全双工方式,所以在实际的系统中必须对接收和发送同时进行处理。

3.2.2 系统维护模块 主要用来处理系统维护信息,主要包括:由管理用户发出的搜索广播报文及参数配置报文。这些报文用于对子网上的所有 DAS 进行搜索和配置。由于广播报文的内容和格式为内部约定,只有网络中的 DAS 设备能够对该信息进行正确的识别及应答,而其它主机设备在正常情况下并不会处理这些报文。管理用户则根据收到的应答数据报文来确定需配置的对象及相关参数。为了保证更新命令的成功操作,DAS 系统在系统更新成功后将返回成功信息给管理用户。系统配置所涉及的参数如表 1 所示,能满足绝大多数串的接口设备的工作要求。

传输。这两个参数由用户设定,如表 1 所示。

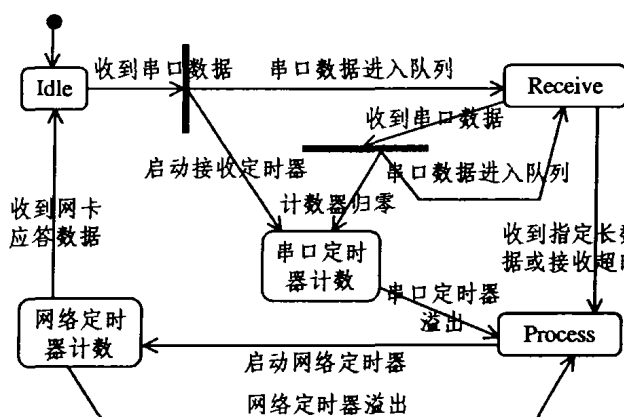


图5 串口接收处理活动图

网络定时器用于检测网络应答是否超时。系统在发出一个数据报文后启动该定时器,在定时器运行期间内,如果收到接收方的应答,则表明上次发送的数据报文已经成功到达接收方。这时停止定时器,进行下一个报文的处理。如果定时器溢出,表明上次发送失败,系统将进行上次数据的重传。为了避免系统无限次重传,定义超过 3 次则表明由于网络原因或接收端原因,使数据传输失败。此时对该报文作丢弃处理。

4.2 串行接口缓冲环

由于串口的速度和以太网的速度差别较大,为了平滑两个接口的传输速度,设计了两个环形缓冲队列分别针对接收和发送进行处理。这个环形缓冲区设置了 8192 字节,起始地址分别为 0x2000 和 0x4000,最多可以存储 8191 个字节的数

(下转第 52 页)

ftp://ftp.ee.lbl.gov/papers/sacks.ps.Z

- 8 Floyd S, Henderson T. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 2582, April 1999
- 9 Jitendra P, Victor F, Don T, Jim K. Modeling TCP Throughput:

A Simple Model and its Empirical Validation. In: Proc. of the ACM SIGCOMM '98 conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication

- 10 陆传贵. 工程系统中的随机过程. 电子工业出版社, 2000. 72~73

(上接第 36 页)

据。若使用 9600bps 的通信速率, 以每个字节占用 10bit 计算, 则该缓冲区可以存储约 8.53s 的连续串口数据。若串口速率为 115200bps 则能存储约 711ms 的连续数据; 从另一个方面说, 如果我们定义向网络发送的最大业务数据报文长度为 1k 字节, 那么该缓冲环可以保留 8 帧完整的数据。当缓冲区中的数据满足条件时, 处理器才进行相应的操作流程。这使得处理器在处理串口数据时能享有更大的弹性, 并且将运算资源用于其它的工作。

4.3 SDTP 协议

图 6 描述了在本 DAS 服务器中实现的 SDTP (Serial Data Transmit Protocol) 协议格式。所有针对数据传输的报文均按照该协议结构进行封装。SDTP 的报文以 1 字节长度的类型字段开始, 包括了 2 字节的序号字段, 2 字节的长度字段和不定长度的数据字段。

类型	序号	长度	数据
1 字节	2 字节	2 字节	可变

图 6 SDTP 协议结构

SDTP 序号字段与长度字段的使用能保证数据的有序传输, 以及通过对称重传机制来防止报文的丢失。

5 性能分析

我们主要分析 DAS 系统对串口数据的处理能力。为了简化计算做如下假设: 设串口的波特率为 B 。传输每个字节需要 10bit, 则通过串口每秒能传输 $B/10$ 字节。根据本文设置的串口传输规则, 数据报文最大长度设定为 L 字节, 等待超时值设定为 D 秒。那么, 系统启动传输的两个条件的最短延迟时间分别为:

条件 1: 收到 L 字节时发送, 延迟为: $10 * L/B$;

条件 2: 超时 D 秒发送, 延迟为: $10/B + D$ 。

系统的最短延迟时间为上述两个参数的最小值, 即 $\min(10 * L/B, 10/B + D)$ 。

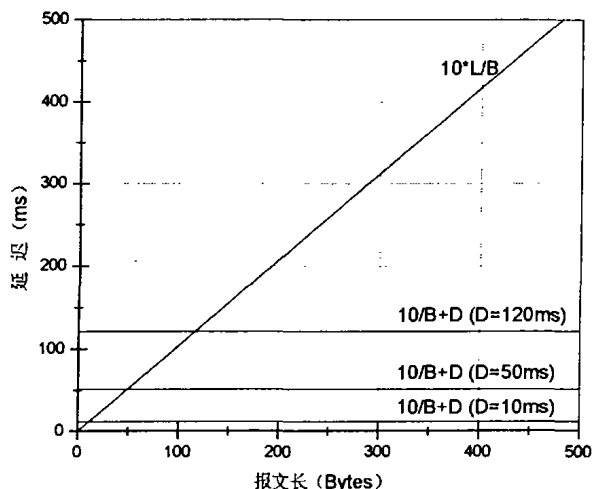


图 7 不同参数下的系统最小延迟时间

考虑最坏情况下的延迟时间: 系统在收到第一个字符后, 每次在超时前的一瞬间收到了串口来的下一个字符, 那么整个数据帧的延迟为 $(D - 10/B) * (L - 1)$ 。

图 7 描述了系统在 9600bps 波特率下, 不同参数设置对系统最小延迟的影响。从图 7 中的曲线可以看出: 对于条件 1, 系统的最小延迟是随报文长度增加而线性增加的; 对于条件 2, 系统的最小延迟为固定值。由于系统总最小延迟为二者的最小值, 因此可以在波特率一定的情况下通过确定 D 的值来确定系统的最小延迟不大于某个值。

同时, 我们针对 Moxa 公司的 NPort Server 产品和 Digi International 公司的 PortServer TS 产品, 以及本文的 DAS 服务器进行了性能比较与测试, 结果如表 2 所示。

表 2 三种产品的比较

	DAS	Moxa	Digi
处理器	8 位 AVR	32 位 ARM	16 位 x86
网络延迟	820us	270us	530us
串口速率	2400-115200bps	300-1.5Mbps	300-921Kbps
网络接口	10M	10/100M	10M
价格	¥100	¥2400	¥1000

通过比较, 可以看出, DAS 与别的产品相比虽然在功能和性能的绝对值上都不及同类产品, 但是, 其所提供的性能已经能够满足很多现场设备的运行要求, 并且在价格因素的比较中具有很大的优势, 能够占有一定的市场份额。

结论 DAS 通用 Internet 设备接入服务器无需对现有设备做任何修改, 不依赖于串行设备自身数据帧格式的结构, “透明”地实现了面向流传输方式的串行接口和面向包传输方式的网络接口之间的数据转换。该研究成果已应用在东大新业的嵌入式 Internet 系列产品中, 部分产品获国家实用新型专利, 并通过了辽宁省科委组织的科学技术成果鉴定。

参考文献

- 1 Lee B H. Embedded Internet System: Poised for Takeoff[J]. IEEE Internet Computing, May 1998. 29
- 2 Agranat I. Embedded web servers in network devices[J]. Communication Systems Design, March 1998. 30~36
- 3 Wolf W. Hardware-software co-design of embedded systems. Proc. of the IEEE, 1994, 82(7): 967~989
- 4 Robert F. Embedded Internet systems come home[J]. IEEE Internet Computing, 2001, 40(14): 52~53
- 5 Jacek W. Embedded Internet technology in process control devices [J]. IEEE, 2000, 34(3): 301~308
- 6 赵海, 陈飞鸣. Embedded Internet 的体系结构及其 ONDC 模型的实现[J]. 东北大学学报, 1999, 20(3): 257
- 7 张德干, 郝先臣, 赵海. 一种基于 EI 技术的 EIDI 模型的研究及实现[J]. 电子学报, 2002, 30(5): 749
- 8 金欢, 阮冠春, 徐凌宇, 赵海. 基于嵌入式 Internet 技术的 Webit 体系结构研究与实现[J]. 控制与决策, 2002, 17(5): 541