

浅析远程监测系统中的数据传输与数据管理

江玉玲

(集美大学 诚毅学院, 福建 厦门 361021)

摘要: 介绍由 NI 公司 LabVIEW 编程的基于 TCP/IP 协议的远程监测系统的数据传输, 利用 SQL - SERVER 对上传的数据进行管理, 简单介绍了 LabVIEW 与数据库之间的连接工具——LabSQL。

关键词: LabVIEW; TCP/IP 协议; SQL-SERVER; LabSQL

中图分类号: TP39

文献标识码: B

文章编号: 1006 - 2394 (2009) 08 - 0043 - 03

Data Transmission and Management of Long-distance Monitoring System

JIANG Yu-ling

(Chengyi College, Jimei University, Xiamen 361021, China)

Abstract: The data transmission of long-distance monitoring system based on TCP/IP protocol programmed by LabVIEW of NI Company is introduced in this paper. The SQL-SERVER is used to manage received data. Also, the connection tool between LabVIEW and database—LabSQL is simply described.

Key words: LabVIEW; TCP/IP protocol; SQL-SERVER; LabSQL

0 引言

LabVIEW 是由美国 NI 公司开发的基于图形化编程语言 G 的开发环境, 它是使用工程技术人员熟悉的术语、图标、概念、图形化的符号编程的一种方法。本文主要介绍基于 LabVIEW 平台, 采用 Socket 网络套接字构建 TCP 协议层的远程监测系统服务器端应用软件; 采用 SQL-Server 实现对监测上传的数据进行管理。系统总体结构图如图 1 所示。

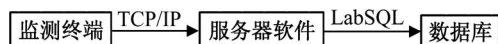


图 1 系统总体结构

1 以太网通信

基于 LabVIEW 平台的 DataSocket 技术, 通过以太网将远程客户端的应用程序以网络数据包的形式传输到远程服务器端。

1.1 TCP/IP 整体构架概述

TCP/IP 协议并不完全符合 OSI 的 7 层参考模型。传统的开放式系统互连参考模型, 是一种通信协议的 7 层抽象的参考模型, 其中每一层执行某一特定任务。该模型的目的是使各种硬件在相同的层次上相互通信。这 7 层是: 物理层、数据链路层、网络层、传输层、

会话层、表示层和应用层。而 TCP/IP 通信协议采用了 4 层的层级结构, 如图 2 TCP/IP 协议簇的 4 个层次, 每一层都呼叫它的下一层所提供的网络来完成自己的需求。这 4 层分别为:

应用层	Telnet, FTP 和 e-mail
传输层	TCP 和 UDP
网络层	IP, ICMP 和 IGMP
链路层	设备驱动程序及接口卡

图 2 TCP/IP 协议簇的 4 个层次

应用层 应用程序间沟通的层, 如网络远程访问协议 (Telnet)、文件传输协议 (FTP)、电子邮件传输 (e-mail) 等。

传输层 在此层中, 它提供了节点间的数据传送, 应用程序之间的通信服务。主要功能是数据格式化、数据确认和丢失重传等, 如传输控制协议 (TCP)、用户数据包协议 (UDP) 等。TCP 和 UDP 给数据包加入传输数据并把它传输到下一层中, 这一层负责传送数据, 并且确定数据已被送达并接收。

网络层 负责提供基本的数据封包传送功能, 让每一块数据包都能够到达目的主机 (但不检查是否被正确接收), 如网际协议 (IP)。

链路层 有时也称作数据链路层或网络接口层, 通常包括操作系统中的设备驱动程序和计算机中对应

收稿日期: 2009 - 05

作者简介: 江玉玲 (1983—), 女, 助理实验师, 在读硕士, 研究方向为通信导航, 航海教育研究。

的网络接口卡,它们一起处理与电缆(或其他任何传输媒介)的物理接口细节。

1.2 TCP/IP的工作原理

TCP/IP的工作原理如图3,在源主机上应用层将一串字节流传给传输层;传输层将字节流分成TCP段,加上TCP包头交给IP层;IP层生成一个包,将TCP段放入其数据域,并加上源和目的主机的IP地址后,交给网络接口层,再交数据链路层;数据链路层在其帧的数据部分装上IP包,发往目的主机或IP路由器处理。在目的主机处,数据链路层将数据链路层帧头去掉,将IP包交给网络接口层再交IP层,IP层检查IP包头,如果包头中的检查和计算出来的不一致,则丢弃该包;如果检查一致,IP层去掉IP头,将TCP段交给TCP层,TCP层检查顺序号来判断是否为正确的TCP段;TCP层检查TCP包头,如果不正确就抛弃,若正确就向主机发送确认;目的主机在传输层去掉TCP头,将字节流传给应用程序。

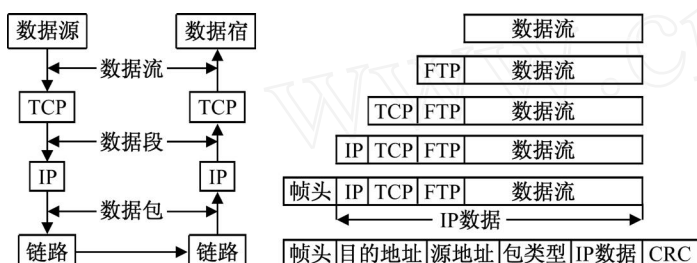


图3 TCP/IP的工作原理

1.3 基于LabVIEW的TCP通信编程实例

LabVIEW可以直接使用TCP/IP协议的传输层,这一层的功能主要是提供应用程序间的通信,TCP/IP协议簇在这一层的协议有TCP和UDP,LabVIEW提供了与这两个协议相关的功能函数。UDP协议较为简

单,但可靠性稍差;TCP协议提供的是面向连接、可靠的字节流服务。当客户和服务器彼此交换数据前,必须先在双方之间建立一个TCP连接,之后才能传输数据。TCP提供超时重发、丢弃重复数据、检验数据、流量控制等功能,保证数据能从一端传到另一端,因而,多数情况下,为了实现可靠性数据传输,考虑使用TCP协议构建PC端服务器。

本模块应用Socket技术构建基于TCP协议的服务器,实现监测终端与远程服务器数据的传输和共享,保证数据的实时性;应用TCP/IP技术实现监测终端和服务器进行对话、数据传输等其余的网络通信功能。

在LabVIEW中可以利用已发布的TCP V I及相关子例程设计实现TCP通信。此前需要在通信的计算机上正确安装TCP/IP协议簇。在已接入Internet/Intranet的Windows/Unix系统中,TCP/IP协议是内置的;若是新接入的计算机系统,须确保TCP/IP正确的安装和设置,同时保证通信所需的源端口/目标端口开放。

由于需要把各数据点的数据汇总到处理主机,设计上采用服务器/客户端通信模式,下面以一个简单的服务器与客户端之间收发的例子来模拟,设计其工作模式如图4,具体服务器端程序如图5,客户端程序如图6。

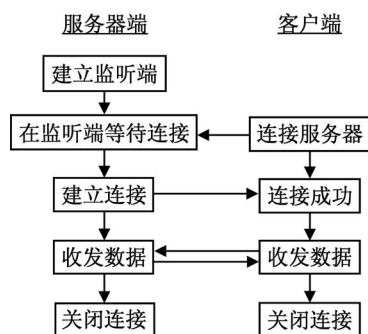


图4 双机TCP通信流程图

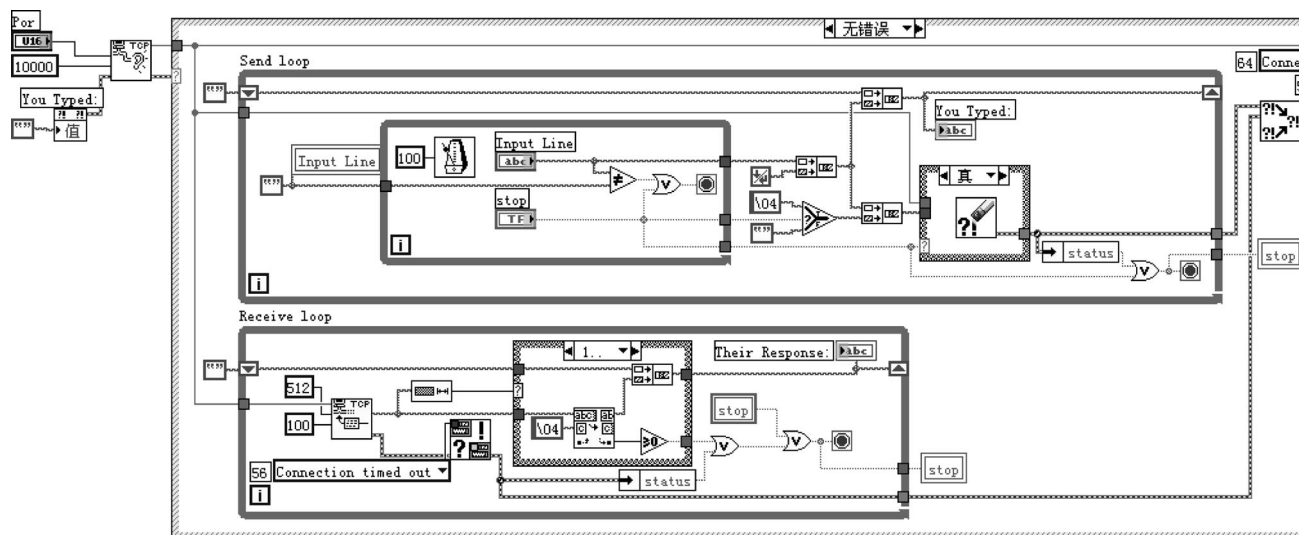


图5 服务器端软件设计

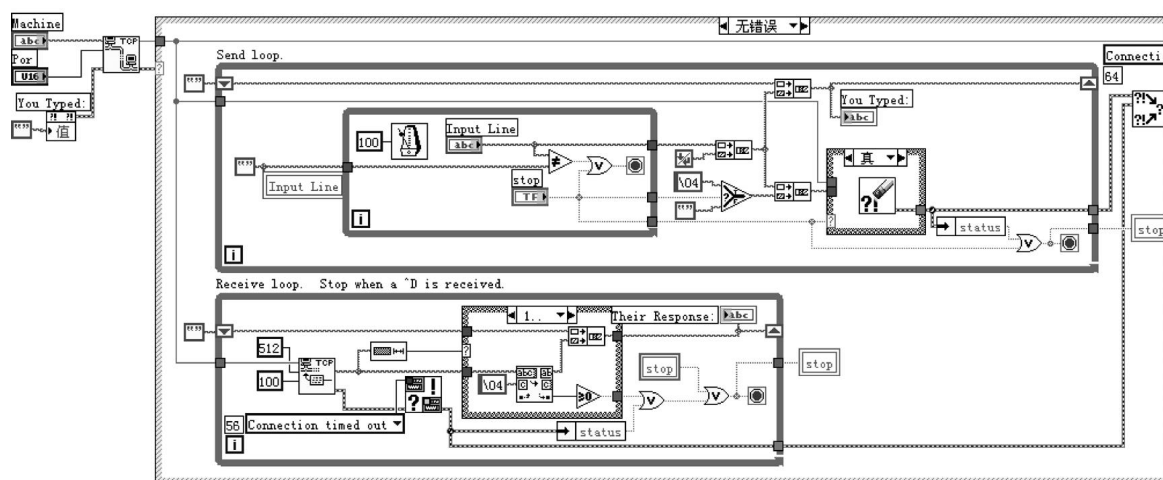


图 6 客户端软件设计

运行仿真结果如图 7 所示,由图中可看出在服务器端软件中输入 abc,则在客户端接收到 abc;同时客户端输入 123,则在服务器端接收到 123,实现了客户端与服务器端的数据传送功能。

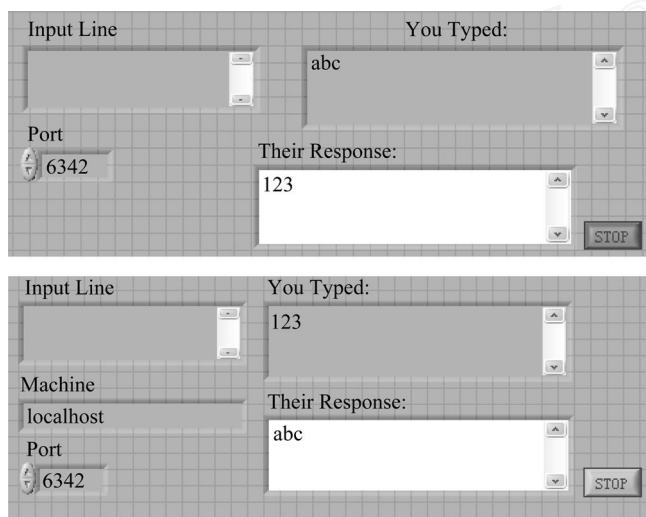


图 7 双机 TCP通信仿真结果

2 LabVIEW 与数据库的连接技术

2.1 数据库访问的工具包 LabSQL 简介

LabSQL 是一个免费的、多数据库、跨平台的 LabVIEW 数据库访问工具包。目前的版本是 LabSQL Release 1.1。LabSQL 支持 Windows 操作系统中任何基于 ODBC 的数据库,包括 Access、SQL-Server、Oracle、Pervasive、Sybase 等。LabSQL 是利用 Microsoft ADO 以及 SQL 语言来完成数据库访问,将复杂的底层 ADO 及 SQL 操作封装成一系列的 LabSQL VIs。LabSQL VIs 按功能可分为 4 类:

- (1) Command VIs, 完成一系列的基本 ADO 操作。
- (2) Connection VIs, 管理 LabVIEW 与数据库之间的连

接。(3) Recordset VIs, 对数据库中记录进行各种操作。

(4) 高层 VIs, 对前 3 类 VIs 进一步进行功能的封装。

2.2 建立 LabVIEW 与数据库的连接

SQL Server 2005 是微软推出的一款基于应用客户机/服务器体系结构的关系数据库管理系统,它提供了较为全面的服务。本文采用 SQL Server 2005 关系数据库管理系统,具体操作步骤如下:

(1) 安装好 SQL Server 2005 后,在对象资源管理器中选择要创建的服务器(这里为本地计算机),右击“数据库”,在弹出菜单中选择“新建数据库”命令,在弹出窗口中,设置数据库的名称“radiation test”,设定数据文件和日志文件的属性。

(2) 将 DAQ 添加到系统数据源 DSN 中,并选择该系统数据源的驱动程序。本文所选驱动程序为 SQL Server,任何具有权限的用户都可以访问该数据源。

(3) 利用 create 命令创建数据表。这里需要建立 4 个表格,分别用来存储远程监测终端上传的分钟数据、小时数据、报警数据和日志文件。打开一个 Connection 对象并调用 Open 方法连接数据源。在 ADO DB Connection 的 Open 方法中需要设置一些与数据库连接的参数,连接字符串为具体的数据源(ODBC),在此为“radiation test”;然后调用 Connection 对象的 Execute 方法执行 SQL 语句,SQL 语句输入可通过软件里的 Command Text 输入,提交给 Execute 方法执行。

2.3 LabVIEW 访问数据库流程

利用 LabSQL 访问数据库的步骤大致分为 4 步:

- (1) 通过 ADO Create vi 创建一个 Connection 对象。
- (2) 利用 ADO Connection Open vi 建立与数据库的连接。
- (3) 利用 ADO SQL Execute vi 执行对数据库的操作(查询、添加、删除、修改)。
- (4) 利用 ADO Connection Close vi 和 ADO Connection Destroy vi 关闭与数据库之间的连接。

(下转第 48 页)

00 5位;01 6位;10 7位;11 8位

线路状态寄存器 (LSR)提供了与 CPU 进行通信过程中的状态信息,其中包括接收寄存器中数据是否已准备好、是否发生数据溢出或奇偶校验错误、发送寄存器是否空等标志,CPU 可在任何时候读取这些状态,以查询的方式与 16C550 进行通信。为了减少 CPU 对接收和发送寄存器读写的次数,节省时间,ST16C550 还提供了 FIFO 模式,FIFO 缓冲区的大小由 FIFO 控制寄存器 (FCR)决定,最大为 14 个字节。

具体的程序代码如下:

```
lcr_dlab. set 80h          ;DLAB = 1,写波特率发生器
baut_dll. set 60h          ;K = 1200kbit/s
baut_dlm. set 00h
lcr_data. set 03h          ;parity = 0, stop = 1, data = 8
ier_data. set 06h          ;使能接收和发送的中断控制位
temp. usect " temp", 5     ;定义 5 个临时地址
initial:
    STM #temp, AR1
    STM #lcr_dlab, *AR1 + ;将初始值放到临时地址中
    STM #baut_dll, *AR1 +
    STM #baut_dlm, *AR1 +
    STM #lcr_data, *AR1 +
    STM #ier_data, *AR1 +
    STM #temp, AR1
    PORTW *AR1 +, 03h ;将初始值写入到 ST16C550 相应的寄存器中
    PORTW *AR1 +, 0h
    PORTW *AR1 +, 01h
    PORTW *AR1 +, 03h
    PORTW *AR1 +, 01h
    ANDM #0, IFR          ;清除中断标志寄存器
    STM #03h, MR           ;打开 NT0, NT1 中断
    STM #0220h, PMST       ;设置中断向量指针 IPTR
    RSX N TM               ;开放总中断
    DLE                    ;等待中断
```

3.2 中断服务程序

中断服务程序分为接收程序和发送程序,将接收的数据存放在 receive_buffer 开始的缓冲区;要发送的数据存放在 transmit_buffer 开始的缓冲区。

程序代码如下:

```
STM #receive_buffer, AR2 ;将接收缓冲区首地址赋予辅助寄存器 AR2
STM #transmit_buffer, AR3 ;将发送缓冲区首地址赋予辅助寄存器 AR3
receive:
    PSHM ST1
    PORTR 00h, *ar2 + ;读入来自接收缓冲寄存器的数据
    POIM ST1
    RETE                ;开中断返回
transmit:
    PSHM ST1
    PORTW *ar3 +, 00h ;将数据写入到发送缓冲寄存器
    POIM ST1
    RETE                ;开中断返回
```

3.3 功能验证

ST16C550 提供了一种 LOOPBACK 模式,将 MO-

DEM 控制寄存器中的第 4 位置 1 进入这种模式。在 LOOPBACK 模式中,RX 和 TX 引脚在内部连接起来,发送的数据经 RX 引脚又接收进来。通过这种模式,可以验证软件设计的正确性。

4 结论

借助串口调试助手,将 PC 机的波特率设置成 1200kbit/s,然后向 ST16C550 发送异步串行数据,变成同步数据后送入 DSP 中。DSP 将接收到的数据再返回 PC 机中,结果显示与发送的数据完全一致,验证了本系统设计的正确性。

参考文献:

- [1] 邓勇,施文康. TMS320C50 与 PC 机高速串行通信的实现 [J]. 计算机自动测量与控制, 2000, 8(5): 23 - 25
- [2] 雷平,缪玲娟,沈军. TMS320VC33 扩展异步串口及串口通信的实现 [J]. 微计算机信息, 2006(2): 140 - 142

(许雪军编发)

(上接第 45 页)

LabVIEW 访问数据库的流程如图 8 所示。

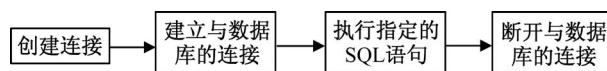


图 8 LabVIEW IEW 访问数据库的流程

3 结束语

本文简单介绍了远程监测系统服务器端软件的构建,包括基于 TCP/IP 协议的数据通信与数据管理。其中,数据通信介绍的是双机通信的一个简单例子,在实际应用中,还可以根据需要扩展为多机通信。而 LabSQL 是 LabVIEW IEW 与数据库之间进行连接的一个工具包,在实际应用中,可以利用其提供的模块,根据实际需要进行数据库中表格的建立,插入记录,以及查询表格等操作。

参考文献:

- [1] 吴松涛,龚家伟. 在 LabVIEW IEW 中利用 LabSQL 实现数据库访问 [J]. 国外电子测量技术, 2006, 25(4): 53 - 56
- [2] 宋晓庆,杨维翰,潘小波. 基于 LabVIEW IEW 的数据库访问技术在电网谐波分析系统中的应用 [J]. 仪表技术, 2007(4): 20 - 25
- [3] 杨乐平,李海涛,杨磊. LabVIEW IEW 高级程序设计 [M]. 北京:清华大学出版社, 2003
- [4] 陈克力. SQL Server 2005 编程基础 [M]. 北京:清华大学出版社, 2008

(丁云编发)