

基于 Socket 通讯技术的上层监控软件的实现

于涛,王健

(东南大学 计算机科学与工程学院,江苏 南京 210096)

摘 要:结合‘计算机板件检测设备’项目,为达到监控下位机的目的,开发了一套基于 Socket 通信技术的实时监控软件。运用面向对象的分析方法对软件框架进行划分,通过继承 MFC 的 Dialog 类进行实现,并针对具体应用详细讨论了通信协议的设计。根据项目特点并参考实时数据库压缩技术提出了接收数据结构的改进,获得了良好的效果。

关键词:面向对象;Socket 通信;MFC;通信协议;数据保存

中图分类号:TP302.1

文献标识码:A

文章编号:1673-629X(2009)03-0243-03

Implementation of Upper - Layer Monitoring Software Based on Socket

YU Tao, WANG Jian

(School of Computer Science & Engineering, Southeast University, Nanjing 210096, China)

Abstract: Introduces how to develop a real-time monitor software based on the Socket protocol for the purpose of monitoring the lower-layer computer which is used in the project of the piece of detecting equipment on computer board. It makes partition on the software frame in object-oriented method, inherits the MFC Dialog class to implement it and discusses in detail the design of the communication protocol. Furthermore, a receiving data structure is proposed according to the project characteristics and the arithmetic of compression and obtains good results.

Key words: object-oriented; Socket; MFC; communication-protocol; data-save

0 引言

PC104 总线是专门为嵌入式控制而定义的工业控制总线。根据‘计算机板件检测设备’项目要求,为检验及检测下位机 PC104 总线接口的扩展模块(DMM-32-AT, EMM-4-XT, QMM-5-XT),设计并实现了一套基于 Socket 通信技术的上层控制端软件。该监控系统能够显示各模块及其接口通道的工作状态信息,为快速、准确地判断各模块状态提供依据。根据项目要求设计了一套用于上位机与下位机交互的通信协议,并引入了一种优化内存存储空间的数据结构,当数据连续一致时可以在原有的存储空间内存储更多的数据,从而可节省内存使用空间,提高了软件监控的可靠性、稳定性。

1 网络编程

Microsoft Visual C++ 的 MFC 类库提供了两个

Socket 套接字类: CAsyncSocket 和 CSocket^[1]。它们由 CObject 派生而来。CAsyncSocket 类封装了 Windows Socket API 函数, CSocket 类派生于 CAsyncSocket。CAsyncSocket 封装了 Windows Sockets API, 并将与 Socket 有关的 Windows 消息转换为回调函数, 使得其更加面向底层, 因而使用起来更灵活。

该软件使用非阻塞式 Socket^[2], 采用客户端连接方式, 在输入服务器端的 IP 地址和端口号之后与下位机的服务器端进行连接^[3,4]。

2 软件设计

监控软件的设计思想是利用 Socket 和 Windows 的消息映射机制以及通信协议, 实现上位机和下位机的控制、反馈以及数据传输, 达到检测和控制的目的^[5]。

该软件的开发使用了以下的方法和技巧: 面向对象的设计思想; 通信协议的设计与实现; 添加消息处理实现多种功能。

2.1 面向对象设计思想

该软件的设计使用了面向对象的编程思想, 基于

收稿日期: 2008-06-11

作者简介: 于涛(1983-), 男, 硕士研究生, 研究方向为嵌入式系统; 王健, 副教授, 研究方向为嵌入式系统、计算机体系结构。

MFC 的 Dialog 类,将任务按模块主要划分为:通信模块、窗口模块、波形显示模块,以及 DMM、EMM、QMM 三个检测信息模块以及控制模块。

通信模块:实现服务器和客户端的连接,数据的接收和发送。

窗口模块:实现人机互动的操作以及显示下位机发送过来的信息。

波形显示模块:在主窗口中动态显示 DMM 板的 A/D 采集数据波形。

D/E/QMM 检测信息模块:封装检测板的检测信息。

控制模块:实现对下位机的控制,主要包括命令的发送、接收以及电压数据的保存、波形的控制(清除、回放)等功能。

各模块关系如图 1 所示。

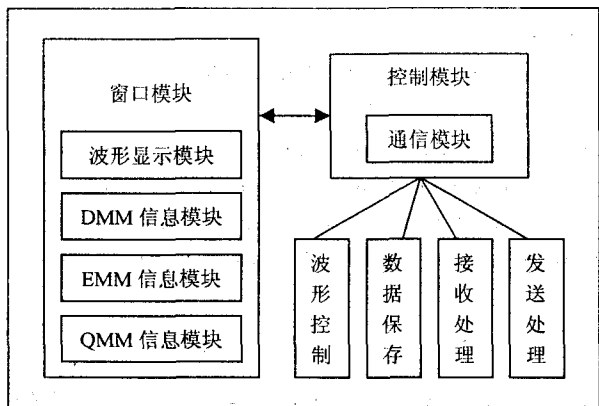


图 1 各模块之间的联系

2.2 通信协议设计

本检测软件用于检测基于 PC104 总线的 DMM、EMM 和 QMM 检测板。根据检测时需要对其进行配置以及数据的传送,通信协议可分为两大类,一类用于对检测板的配置和反馈的命令通道;另一类则用于接收和发送纯数据的数据通道^[6]。

网络中一次数据的发送和接收的长度受到 TCP/IP 协议的限制^[7],在数据量比较大的时候将一组数据分多次发送。本协议规定一次发送和接收帧的最大长度是 4096 个字节(byte0~byte4095),由于前 7 个字节的内容固定,这时就有三种情况^[8]:

- 1)当纯数据长度小于或者等于 4089 字节的时候,一帧 $4096 = 7 + 4089$,所以只有首帧。
- 2)当纯数据长度大于 4089 字节并且小于或等于 8185 字节的时候,两帧 $4096 \times 2 = 7 + 4089 + 4096 = 7 + 8185$,所以由首帧和尾帧构成。
- 3)当纯数据长度大于 8185 字节(即长度超过两帧 $4096 \times 2 - 7 = 8185$)的时候,首帧和尾帧之间夹杂着长

度固定为 4096 字节的帧。

从简化程序设计的角度出发,设定通信协议的前 7 个 Byte 为固定格式,如表 1 所示。

表 1 协议前 7 个字节的内容

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
数	据	长	度	命令号	紧急标识	扩展
数据长度最高 8 位	数据长度次高 8 位	数据长度次低 8 位	数据长度最低 8 位	0	1	

前四个字节为此次通信过程的数据总长度,目的使接收方能够确定接收数据总量便于分配接收空间;其后为一个字节的控制信息,称为命令号,接收方根据此信息对后面的接收内容进行处理;第六个字节为紧急标识位,由于下位机的通讯使用双缓冲机制,默认采用 FIFO 通讯方式,使用此信息便于处理紧急消息增强交互的实时性。Byte6 为用于以后协议的扩展升级而增加的保留位。

从 Byte7 开始的协议内容要根据具体应用而设定,其特点是协议长度固定,协议内容确定,每个字节对应一个参数设定。下面以 A/D 采样的参数配置协议为例,说明命令通道的协议内容(见表 2)。

表 2 具体协议的内容

Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
扫描类型	通道	选择	扫描间隔	测量范围	模式
端口扫描模式。0 为 scan 模式;1 为 single 模式	0-31 标识低端通道号	0-31 标识高端通道号	0-3 分别标识 5μs 10μs 15μs 20μs	0-2 标识电压为 0V ±5V ±8.5V	0 为单极;1 为差分

相对命令通道的数据通道的协议,其特点是协议长度不确定,对应的数据所占长度需要根据具体应用而确定,以 A/D 采样的数据发送协议为例,如表 3 所示。

表 3 协议分段

首帧(前 7 位同上)

Byte7	Byte8	Byte9	Byte10	Byte11	...	Byte4095
通道	一个	接收	到的	数据		
通道号	四	个	字	节		

中间帧

纯数据(长度等于 4096byte)

尾帧

纯数据(长度小于或等于 4096byte)

通信协议的使用可将待发的数据组合成一个整体的数据包,便于发送和接收。这样可以有效降低数据发送次数,减轻对处理资源的占用并可以减少网络数据的传输量^[5]。

2.3 添加消息处理

Windows 程序是基于事件驱动的编程模式,应用程序大部分是响应 Windows 的消息而进行操作的。在本软件的实现中涉及了大量的系统消息处理和添加

的消息响应。

应用程序所要做的每项工作几乎都是基于处理 Windows 消息的,这些消息以三种基本形式出现:常用的 Windows 消息、控件响应和命令。

由于是通过使用 MFC 中继承 Dialog 类来实现,大部分的消息映射已由系统所提供。可以根据需要在 ClassWizard 中进行添加。具体的不予赘述。

3 重要数据结构

DMM 检测板数据采样速率最大可达 200 000 次/秒。因此如何有效处理接收数据是影响软件运行的一个关键因素。

在 MFC 的 CAsyncSocket 类中,每当有数据发送过来时,会通过消息 OnReceive(int nErrorCode)^[2]通知在系统的数据接收缓冲区中存有数据,然后可调用 Socket 的接收函数 Receive(pBuff, remainLen, 0)将数据暂存入 pBuff 中。pBuff 仅暂时保存接收数据,必须在下次接收数据前将其中的数据保存到指定的位置,用于上层监控软件的处理。

实际中下位机每隔 0.2 秒对 A/D 端口进行扫描,并将采集到的数据一次性发送过来,上位机软件根据协议前四个字节的内容创建对应空间并将 pBuff 中的数据进行保存。

参考实时数据库中的死区限值压缩算法^[9](判断当前值偏离最后一个记录的范围是否大于死区限值,决定是否记录此数据。)并根据 DMM 检测板 AD 采集到的数据在很长时间内相对固定这一特点,在本软件的实现中采用以下自定义的链表结构,其结构如下所示:

```
struct SaveBuff
```

```
{
    SaveBuff * pNext;
    float y;
    int nu;
};
```

pNext 指针指向下一个数据存储位置;y 值保存接收数据;nu 用于标识此数据的个数。在接收时,申请一个数据的存储空间,存储接收的数据值到 y,同时 nu 标识为 1。接收下一个数据时,首先判断当前接收数据是否与前一个数据相同,如果相同则不必分配存储空间而将前一个数据结构中的 nu 加 1。只有当先后的两个数据不同或 nu 值大于 X(本软件设计 X 为 100)时,才分配一个新的存储结构。此数据结构是死区限值为 0 的压缩算法的实现^[10],但不同的是此方法在前后数据一致的情况下,即可以保存所有接收到的

电压数据又能够实现节约内存空间的特点。

测试中,最坏情况下(也就是前后数据均不相同时),其存储效率、分配空间与普通链表结构相同;在连续数值相同的情况下,其性能优于普通链表结构。如图 2 所示,测试数据为 200000 个 float 型数据,共占 800000bytes 约 785k,循环发送 10 次。在此情况下通过 Windows 任务管理器所示,上面的 WinClt 程序使用的是普通链表结构(也就是最坏情况),对照的是下面的 WinClt 程序,则是使用 SaveBuff 数据结构链表。

映像名称	用户名	CPU	内存使用
vsnpstd3.exe	yu_tao	00	4,360 K
Server.exe	yu_tao	00	5,584 K
WinClt.exe	yu_tao	05	27,512 K
taskmgr.exe	yu_tao	02	2,860 K
SynTPlpr.exe	yu_tao	00	2,604 K

映像名称	用户名	CPU	内存使用
mspaint.exe	yu_tao	00	1,688 K
WinClt.exe	yu_tao	05	4,352 K
SynTPlpr.exe	yu_tao	00	2,604 K
Xming.exe	yu_tao	00	15,036 K
LTSMSG.exe	yu_tao	00	2,848 K
menu.exe	yu_tao	00	3,176 K

图2 两种结构运行对比图

4 结束语

以 Socket 技术为背景,研究并实现了上位机与下位机实时监控软件的设计与开发,着重讨论了通信协议的设计模式以及大数据保存时节省内存空间的问题,并通过试验测试软件的性能。文中提出的通信方式在性能上与传统的方法差别不大,但这种方式便于上层应用程序实时监控下位机状态,提高了应用的可靠性。本检测系统已经投入运行使用。从运行的情况看,客户端和服务端之间的通信比较流畅,能够进行良好的实时检测。

参考文献:

- [1] 潘爱民,王国印. VisualC++ 技术内幕[M]. 第4版. 北京:清华大学出版社,2001:156-230.
- [2] 王华,叶爱亮,祁立学. VisualC++ 6.0 编程实例与技巧[M]. 北京:机械工业出版社,1996:137-216.
- [3] 王恩涛,李祥. 基于 Socket 的手机与数据库服务器通信的研究[J]. 计算机技术与发展,2007,17(2):82-84.
- [4] Canha R M F, de Cresce El Debs A L H, El Debs M K. Design model for socket base connections adjusted from experimental results[J]. Structural Concrete,2007(1):264-277.
- [5] 陈冬梅. 基于 C-S 的实时网络监控软件的开发[D]. 武汉:

(下转第 248 页)

货实际总重、超载极限、计重费率及所属路段各业主所占里程,按上述计重货车通行费计算公式模型,可计算得到各路段业主 e_i 应得通行费为 X_i 元。

(2) 计算实际拆分比例:首先,将各路段业主应得通行费汇总,从而得到该货车经过所有路段的原始通行费收入(精确值)为 X 元: $X = \sum_i X_i$ 。其次,用各业主应得通行费 X_i 除以总的应收通行费 X 计算出每个业主所占比例为 α_i : $\alpha_i = \frac{X_i}{X}$ 。再次,计算的比例为精确值,由于存在小数位的问题,为保证最后比例之和为 1,对于余数的处理办法为通过比较 $\alpha_1, \alpha_2, \dots, \alpha_n$ 的值的大小,假设其中 $\alpha_m (1 \leq m \leq n)$ 最大,则: $\alpha_m = 1 - \alpha_1 - \alpha_2 - \dots - \alpha_n$ 。如果 $\alpha_1, \alpha_2, \dots, \alpha_n$ 中,存在业主 e_{m1} 对应的 α_{m1} 和业主 e_{m2} 对应的 α_{m1} 和 α_{m2} 均为最大,且业主 e_{m1} 的编号大于 e_{m2} ,则: $\alpha_{m1} = 1 - \alpha_1 - \alpha_2 - \dots - \alpha_{m2} - \dots - \alpha_n$ 。

(3) 计算各路段业主最终所得通行费:按计费取整办法,得到该货车实际收取的通行费为 Y 元;用实际收取通行费 Y 乘以业主 e_i 所占比例 α_i ,得到业主 e_i 最终所得通行费为 $Y_i = Y \times \alpha_i$,其中: $Y = \sum_i Y_i$ 。从而得到该车辆经过业主 $e_1, e_2, e_3, \dots, e_m, \dots, e_n$ 的路段后的最终拆分结果为 $Y_1, Y_2, Y_3, \dots, Y_m, \dots, Y_n$ 。业主最终所得通行费是实际收取通行费按各业主所占比例分别拆分后得到的,与前面计算的各业主应得通行费可能存在一定的偏差。

联网收费拆帐系统实现通行费拆分结算的基础是生成拆分标准表,通过基于路径的高速公路联网收费清分方法^[6]可以生成拆分标准表。拆分标准表记录了整个联网路段任意入口站和出口站的里程、收费金额、要经过路段的业主以及按车型每个业主所得通行费金额等信息。实施计重收费后,对于计重货车,按照实时计算各业主所占通行费比例的方式,根据拆分标准表以及上述通行费拆分具体算法,拆分出各业主所得通行费。

在实际使用中,由于该省高速公路计重收费标准完全统一,为减少计算复杂度,设计系统时预先计算好每条路径中各业主所占里程比例。当各路段费率一致时,直接按照各业主所占里程比例进行拆分;当各路段费率不一致时,必须根据各自的费率公式、里程重新计算,考虑到系统的通用性和可扩展性,仍然保留路段费率不同情况下的计算方式,以便今后扩充。

3 结束语

文中提出的通行费计算算法可以根据需要灵活修改计重费率,以满足不同时期的收费标准及治理超限目标的需要。另外,通行费拆分算法有着简单方便的特点,当路网中加入新的路段时,只需修改拆分标准表,而不必修改整个算法,因而快捷方便,并且对已开通过路段的软件系统影响最小。通过在该省高速公路联网计重收费拆帐系统中的应用,说明了该算法的有效性和实用性。

为了保证通行费拆分结算公正、准确和高效,特别是当联网路段出现越来越多的环状(多路径)情况后,能更最大限度地保护高速公路业主的利益,该算法还有待进一步改进和完善。

参考文献:

- [1] 刘伟铭. 高速公路收费系统理论与方法[M]. 北京:人民交通出版社,2004:10-15.
- [2] 高 博,左庆乐,王应敏,等. 关于我国计重收费几个问题的探讨[J]. 交通标准化,2007,1(2):50-54.
- [3] 曾江洪,李 晓,张 欢. 公路计重收费标准的确定方法及应用[J]. 中南公路工程,2005,30(1):154-157.
- [4] GB1589-2004 道路车辆外廓尺寸、轴荷及质量限值[S]. 2004.
- [5] 杨晓林,范 雯,丁露露,等. 湖北省高速公路计重收费拆分方案[J]. 交通与计算机,2006,24(6):88-90.
- [6] 吴岳忠,何频捷,李长云,等. 一种基于路径的高速公路联网收费清分方法[J]. 计算机技术与自动化,2006,25(3):92-94.

(上接第 245 页)

武汉理工大学,2005.

- [6] 程敬原. VxWorks 软件开发项目实例完全解析[M]. 北京:中国电力出版社,2005:56-168.
- [7] 张 雷,梁建武,陈 英. 串行通信协议的研究与实现[J]. 现代计算机,2006(9):57-58.
- [8] Park Young-Min, Yoo Ji-Yoon, Lee Sang-Bin. Practical Implementation of PWM Synchronization and Phase-Shift

Method for Cascaded H-Bridge Multilevel Inverters Based on a Standard Serial Communication Protocol[J]. IEEE Transactions on Industry Applications,2008(1):273-276.

- [9] 徐 慧. 实时数据库中数据压缩算法的研究[D]. 杭州:浙江大学,2006.
- [10] 姜 磊,黄广君. 自适应的无损数据压缩算法[J]. 计算机工程,2008(1):103-105.