

## 摘要

快速锻造液压机是实现自由锻生产精密化和快速化的压力加工设备。该设备采用大流量电液比例阀和计算机技术,实现液压机的闭环控制和实时监测。本课题对构建监测控制系统的一些关键技术问题进行了详细研究,具体开发一种基于虚拟仪表的状态监测与故障诊断系统,全面而详细地介绍了如何运用功能强大的 Visual C++ 进行虚拟仪表可视化软件的设计以及实现串口及网络间数据传输的方法。其主要内容包括:

1) 探讨了在线监测与故障诊断系统的设计思想和软、硬件构成,简要介绍了系统的总体结构,各部分的功能与特点,同时讨论了状态监测软件的设计思想和实现方法。另外,还给出了实现该系统的部分细节技术问题。

2) 介绍了串行通信接口标准及通信原理,研究了在 Win32 环境下实现串行通信的方法以及需要注意的问题。

3) 分析了 TCP/IP 网络协议及采用 Windows socket 进行网络编程的基本原理,同时还介绍了 Windows socket 开发网络通讯应用程序的设计方法和环境要求。

4) 在人机界面的设计方面,贯彻“仪器仪表化”的虚拟仪器可视化思想,采用面向对象技术设计开发了一套虚拟仪器系统通用控件包,全面地实现了“软件即仪器”的设计构想,对同类仪器的开发研制可以提供一定的参考和实用价值。

实践证明,该系统能对设备故障进行诊断和监测,降低故障诊断的难度,缩短故障诊断时间,提高工作效率,提高故障诊断结论的正确性和精确性,方便故障管理。

**关键词:** 快锻机、状态监测、故障诊断、虚拟仪器、可视化、TCP/IP、Socket

## Abstract

High speed forging hydraulic press is one kind of metal forming machine to carry out precise and fast open-die forging. Adopting the electro-hydraulic proportional valve and computer technology, the closed-loop control and real-time supervision of press was realized. In this thesis, some key technologies on how to build the monitoring and control system were researched. A Virtual Instrument based Condition Monitoring and Fault Diagnosis System was designed, and in detail introduces how to design visual software of Virtual Instrument and implements data transfer via serial ports and network with the powerful tool, VC++6.0. The main contents and achievements of the research are as follows:

1) The thesis introduces the design idea of real-time monitoring and fault diagnosis system, including its software and hardware. The structure, function and special features of the system are introduced in detail. The methods of software planning for the condition monitoring is also discussed. Moreover, some details for designing the system are given also.

2) This paper introduces the standard of serial communication interface and principle of serial communication. The methods of serial communication design in Win32 are given. Some problems that need to be noticed are also introduced.

3) This paper introduces the TCP/IP protocol and the principle of Windows socket programming. The designing procedure and environment request for developing windows network communication is also introduced.

4) And on designing the interface, there are some universal controls be developed to make a Virtual Instruments (VI) interface which carries out the idea of *The software is the instruments*. All of these have some merits on reference and practicality for congeners.

Facts have proved that An intelligence diagnosis system was built for diagnosing and

# 华中科技大学硕士学位论文

---

monitoring the hydraulic pump stations which has yielded remarkable economical profits: readiness in diagnosis, reduce diagnosis time, enhanced productivity, improved precision and accuracy of the diagnosis, and facilitated fault control.

**Keywords:** High Speed Forging Press, Condition Monitoring, Fault Diagnostics, Virtual Instrument, Visualization, TCP/IP, Socket

# 华中科技大学硕士学位论文

---

## 1 绪 言

### 1.1 课题概述

#### 1.1.1 课题来源

华中科技大学武汉新威奇科技公司与兰州兰石新技术开发实业公司合作进行 8MN 快速锻造液压机组控制系统研究与开发。其中快锻液压机组的状态监测与故障诊断系统是控制系统的重要组成部分，它的研究与开发也就成为一个重要的课题而被提了出来。

#### 1.1.2 课题提出

在生产领域中，设备维护是保证生产过程在安全、可靠前提下高效运行的重要手段。有效的设备维护不仅要求维修人员比较清楚的了解设备的组成和工作原理，以及具备大量的维修经验，而且还应借助于可靠的监测和分析手段，更多地获取设备内部的信息。这对大型复杂工装设备的维护是十分必要的。本文所涉及的 8MN 快锻液压机是一台大型液压设备。为提高快锻机工作的安全性和可靠性，研制了一套快锻液压机组的状态监测与故障诊断系统，将科学的测试手段和分析方法与维修人员丰富的经验结合起来，使设备的维护工作更趋于完善。

### 1.2 状态监测与故障诊断技术的发展现状

#### 1.2.1 国外设备状态监测和故障诊断技术的发展状况

国外开展机械故障诊断技术已有 40 年的历史。早在 20 世纪 60 年代末，美国宇航局（NASA）倡导成立美国机械故障预防小组（MFBG），英国成立了英国机器保健中心（U.K. Mechanical Health Monitoring Center），开展机械故障诊断技术的研究。此后各国政府和公司广泛成立组织，召开会议开展研究。20 世纪 70 年代，电子测量技术和频谱分析技术被应用到故障诊断中，国外大型机电系统的监测和故障诊断技术开

始进入实用化阶段,但此时的振动监测只能给出早期的报警,还有待于进一步的发展。到了20世纪80年代中期,以计算机为中心的现代机械故障诊断技术得到了迅速发展,涌现出许多商业化的计算机辅助监测和故障诊断系统。如美国 Scientific Atlanta 公司的 M600CHAMMP 系统, Bently 公司的 ADRE 系统和 ENTECK 公司的 PM 系统等。在这一阶段,由于各种传感技术的飞速发展,使得诊断可以利用振动、噪声、温度、力、电、磁、光、射线等多种信号,从而发展了振动诊断技术、声发诊断技术、光谱诊断技术、无损探测诊断技术、热成像监测诊断技术等。同时由于信号分析技术的发展,模式识别技术、模糊数学、灰色系统理论等新的数学处理工具在故障诊断技术中得以应用。至此,以传感器技术、信号分析技术和计算机技术中心的现代机械故障诊断技术已经形成。20世纪80年代中期开始,以信息处理技术为基础的现代机械故障诊断技术开始向基于知识的智能诊断技术方向发展,人工智能理论得到迅速发展,其中专家系统更被广泛应用到各个领域。以信息处理技术为基础的现代机械故障诊断技术开始向基于知识的智能诊断技术方向发展,并陆续涌现出许多商业化的状态监测和故障诊断系统<sup>[1]</sup>。著名的有:

1) 美国西屋公司于20世纪80年代中期推出的过程控制系统 PDS。该系统利用汽轮机专家建立知识库,采用基于规则的正向推理方式。

2) 美国 Bently 公司于1993年推出的工程师帮助软件,具有两级法诊断能力。初级诊断供现场运行人员使用,高级诊断供中试所和其他专业人员使用。

3) 日本三菱重工研制的机械状态监测系统 MHMS 经历了8~10年的研制历程。目前正在为系统配置以规则知识与框架知识相结合的 Master 推理机制。

还有其他一些系统,如美国 IRD 公司的紫宝石(Amethyst)系统,美国 BEI 公司 DATM4 系统,瑞士推出的 Vibro-View 系统,加拿大的 CSI 系统等等。

## 1.2.2 国内设备状态监测和故障诊断技术的发展状况

我国开展机械设备状态监测和故障诊断技术较晚,从发展历程看,大体经历了三个阶段:

第一阶段从20世纪70年代中期到80年代初期,在引进和消化吸收国外先进技



术的基础上,开始研究各种机械设备的故障机理、诊断方法以及简便的监测和诊断仪器等;

第二阶段从 20 世纪 80 年代初期到 80 年代末期,在借鉴国外先进技术的基础上,开始尝试研究各种新的诊断技术,研究和创建新的诊断理论和诊断方法,同时开始在生产中推广应用设备诊断技术;

第三阶段从 20 世纪 80 年代末期至今,从理论上和生产应用上建立较有系统性的诊断理论,研究设备状态监测和故障诊断系统,将我国的机械故障诊断技术推向理论研究和普及应用的新高潮<sup>[1]</sup>。

我国设备故障诊断技术尽管起步较晚,但近年来发展非常迅速。我国从事设备维修工作的部门,在积极学习国外现代设备管理技术的过程中,对设备诊断的重要性逐渐加深了认识。国内许多高校和研究机构联合厂家开展了设备故障诊断技术的理论和应用研究,取得了一定的科研成果。特别是在一些重要的大型液压机组中,设备故障诊断技术渐渐得到应用,并取得了显著的经济效益。通过应用在线监测与故障诊断系统,可将机组置于全过程监控之下,连续监测诊断各种问题,有效保证机组安全运行,为适时维修提供依据。因此对大型机组安装在线监测与故障诊断系统是今后的必然趋势<sup>[2]</sup>。

近年来,设备在线监测系统在冶金、电力、化工等企业得到广泛应用。宝钢、武钢、太钢等钢铁公司在一些关键机组上安装了在线监测系统,马钢三烧结厂 1<sup>#</sup>、2<sup>#</sup>主抽风机,动力厂 1<sup>#</sup>高炉风机安装了在线监测系统,气体销售分公司 DH80、DH90 空压机组和 3.5 万 m<sup>3</sup>空压机组还安装了网络化在线监测系统。2002 年大庆石化公司根据大机组的具体情况特点,吸收国内外大机组状态监测和故障诊断系统的优点,研制了具有高新技术特色适合本企业的大机组监测诊断网络系统。应用该网络系统可避免重大事故的发生,减少维修费用,延长设备安全运行周期,提高大型转动设备管理水平。高新企业广州珠江钢铁有限责任公司(简称珠钢)根据智能诊断系统的技术要求建立了 CSP 轧机液压在线监测系统对设备故障进行诊断和监测,该系统达到 90 年代先进水平,经过 3 年多的设计、安装和调试,于 1999 年 8 月投入使用。珠钢轧机液压在线监测系统采用高性能计算机、先进传感器、以及 LAN 网等先进技术,实现

了系统结构简单清晰、使用方便、人机界面友好、诊断与监测功能丰富、且造价适中的设计目标，为企业带来了巨大的经济效益。抚顺钢厂为进一步提高快锻机工作的安全性和可靠性，使设备的维护工作更趋于完善，研制了一套 2000t 快锻机在线工况监测系统。该系统于 1995 年 10 月完成调试并投入使用，从使用情况来看，该系统在信号检测、数据传递和显示等方面达到了设计要求。此外，在液压水闸启闭机控制系统、工程机械液压伺服控制系统、收割机液压系统、大型往复式压缩机中，设备在线监测与故障诊断装置都是系统中不可缺少的重要组成部分。

## 1.3 课题研究的目及意义

现代机械设备由于受到许多无法避免的因素的影响，会出现各种故障。设备发生故障后不仅会降低或失去其预定的功能，有时甚至造成严重以至灾难性的事故。国内外曾经发生过多起因设备故障而导致的空难、海难、爆炸、断裂、倒塌、毁坏、泄漏等恶性事故。这些严重的灾难性事故不但造成了巨大的经济损失，而且导致了很大的人员伤亡和环境污染，在社会上引起了强烈的反响，例如 1972 年日本夫西电力公司南海电厂 3 号机组——600MW 汽轮发电机组因振动引起严重的断轴毁机事件，1985 年我国大同电厂和 1988 年我国秦岭电厂的 200MW 汽轮发电机组的严重断轴毁机事件，都造成了巨大的经济损失，因此保证设备的安全稳定运行，消除事故，是十分迫切的问题<sup>[3]</sup>。

液压机是利用液体作为工作介质，通过液压缸产生作用力使金属产生塑性变形。常用的自由锻造液压机，由于受传统的自由锻工艺和液压机本身的影响，存在打击次数少，锻件加工余量大等缺陷。随着液压技术和计算机技术的发展，对液压机的改进成为可能，其发展方向是对传统液压机的计算机控制和新型液压机的设计。快速锻造液压机组是自由锻造设备中目前最为先前的，是机电一体化的大型设备。它依靠现代液压控制技术实现高精度的位置控制和快速锻造，从而在自由锻过程中，使锻件在短时间内完成成形。它生产效率高，控制精度好、节能、可大大改善工人的劳动条件及环境<sup>[4][5][6]</sup>。

设备是企业生产和经济发展的物质技术基础，设备的运转效益与企业的经济效益

---

密切相关。快锻液压机组状态监测与故障诊断管理系统能够加强设备运行管理,进行在线监测,及时发现异常情况,加强对故障的诊断与预防,并且根据设备的状态决定是否需要维修,做到视情维修,大大节省投入到设备维护和维修上的人力、物力和财力;同时也最大限度地利用设备,提高设备的运转效益。

液压系统状态监测及故障诊断是目前国内外液压界普遍关注的一个热点和难点问题,这个问题的研究在推动液压设备维修制度的改革,预防突发事件,保证人身和设备的安全,提高设备效益,实现监测诊断自动化和智能化等方面有着十分重要的意义和广阔的工程应用前景。因此,研究开发“快锻液压机组状态监测与故障诊断管理系统”有重要的意义<sup>[7][8]</sup>。

## 1.4 课题主要研究内容

本课题所要研究开发的快锻液压机组状态监测与故障诊断管理系统是快锻液压机组计算机控制系统的一部分。该系统运行在上位机(监测计算机)上。功能包括:动态显示、记录和打印系统的监测参数和状态,完成信号的分析处理与故障诊断。系统的目的就是提高设备的可靠性和利用率,降低设备维修费用,实现设备状态监测和诊断数据的计算机管理,为建立设备正常状态的判别标准和积累运行状态动态信息档案打下基础,为设备的视情维修创造条件<sup>[9]</sup>。

下面列出系统要完成的主要功能及技术:

- 1) 将工控机(下位机)传送过来的压力、位移和温度等参数以曲线及图表的形式实时动态显示,同时显示工控机进行压机和操作机位置闭环控制的控制参数与位移、压力、速度的相关曲线,以便于进行控制参数的调整。
- 2) 屏幕最下一行显示系统的工作状态及故障(InPlcStatus.In 中的内容,应能闪烁变换),同时能记录开机与关机时间,每一故障发生与解除的时间,故障原因等。
- 3) 以表格显示所有传送到监测计算机中的参数。每一月保存一个文件,并能调出显示及打印等。
- 4) 将压机的位移值通过串行口送大字符显示器显示。
- 5) 及时判断设备是否出现异常或发生故障以及故障的严重程度如何,及时停机



华 中 科 技 大 学 硕 士 学 位 论 文

检修，避免严重事故发生。通过设备状态定期监测，获取设备运行状态的特征信息，掌握设备运行状况的变化情况。

在软件开发上, 本系统采用 MFC (Microsoft Foundation Classes) 作为应用程序框架 (Application Framework), 开发工具选用 Visual C++ 6.0, 基本设计思路是采用单文档——多视图程序结构, 即 SDI 程序的扩展, 而数据的接受、处理、保存等操作则全部交给整个程序唯一的文档类 CMonitorDoc 来进行。程序的各种功能是通过从 MFC 中具有相应功能的基类中派生出自定义类的方法来实现。虚拟仪表控件派生自 CStatic 类, 用于信号及参数的直观显示; 仪表盘视图派生自 CFormView 类, 用于按系统需求摆放虚拟仪表控件; 内存设备环境类 CMemDC 派生自 CDC 类, 用于实现动态图形显示中的双缓冲性能; 异步套接字类 CSocketSer 派生自 CAsyncSocket, 用于实现 TCP/IP 协议下的双机互联; 数据记录集类 CDaoRecordsetAccess 派生自 CDaoRecordset 类, 用于操纵故障数据库中的记录。图 1-1 就是整个快锻液压机组状态监测与故障诊断管理系统的类别组织框架图 (Class Hierarchy)。

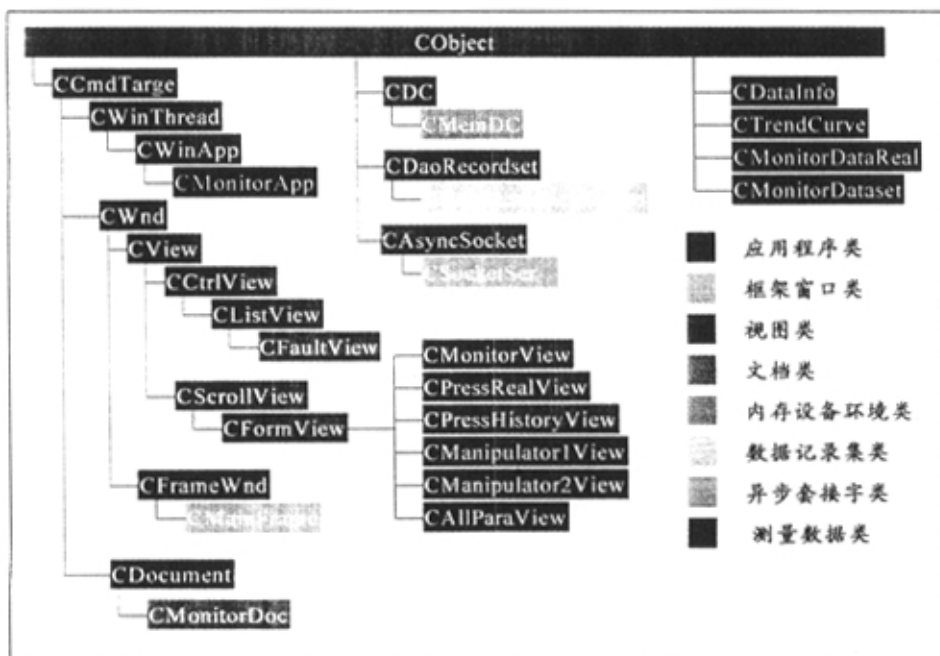


图 1-1 快锻液压机组状态监测与故障诊断管理系统的类别组织框架图

## 1.5 本文的主要工作

本文共分为六章，其研究内容可分为三部分：第一部分针对课题要求完成了快锻液压机组状态监测与故障诊断管理软件系统的总体设计，详细介绍了系统的总体设计方案，所采用的关键技术及预计要达到的功能要求；第二部分对计算机监控系统进行了研究，分析了其软硬件组成及性能特点，最后结合课题完成了快锻液压机组计算机控制系统的设计；第三部分就软件开发中的一些关键技术问题作了具体的分析说明，如通信接口程序的开发、虚拟仪表的实现、故障数据库的建立与管理等。

各章主要内容如下：

第一章是全文的概述，主要介绍了课题的背景、研究的目的意义和主要研究内容，并介绍了国内外的研究概况。

第二章全面介绍了快锻液压机组状态监测与故障诊断管理软件系统的总体设计方案，简要介绍了所采用的主要技术手段——MFC 和面向对象软件开发技术。还分析了软件的结构和主要功能模块，并对每个模块要完成的功能作了详细说明。

第三章对计算机监控系统进行了研究，分析了其软硬件组成及性能特点。最后结合课题完成了快锻液压机组计算机控制系统的设计。

第四章详细讨论了串行通信接口程序的开发和网络通信程序开发的原理及实现，介绍了在 Win32 环境下实现串行通信的方法以及需要注意的问题，分析了 TCP/IP 协议组成以及通过 Socket 接口实现非阻塞方式通信的基本原理及步骤。

第五章对虚拟仪器技术在计算机监测控制系统中的应用作了探讨，介绍了如何在 Visual C++ 中制作虚拟仪表控件及虚拟仪器软面板的方法，以及实现数据动态图形显示的技术关键，最后还对软件中数据结构的设计及数据库的建立与管理作了简要介绍。

第六章是全文总结与展望。

## 2 系统总体设计

### 2.1 总体技术方案

本课题研究开发的快锻液压机组状态监测与故障诊断管理系统是一个基于网络的虚拟仪表系统，主要用于对采集到的数据和状态信号进行实时监测和事后处理。从实用性、研制周期、性能价格比和应用推广等角度，本系统在应用程序框架(Application Framework)上决定采用 MFC (Microsoft Foundation Classes)，开发工具上选用 Visual C++ 6.0，并且运用面向对象(Object Oriented) 技术设计全部系统所要用的虚拟仪表控件，用于信号及参数的直观显示。基本设计思路是采用单文档——多视图程序结构，即 SDI 程序的扩展，单文档指仅包括一份 Document，其特点是开启第二份 Document 时，必须先把第一份 Document 关闭；多视图是指数据可以不同的类型显现出来，并以“同源子窗口”显示代表同一份 Document 而又各自分离的 View 窗口。视图则采用最适合制作虚拟仪表盘的 CFormView 类，将开发出来的虚拟仪表控件按要求摆放到各个 FormView 视图上。而数据的接受、处理、保存等操作则全部交给整个程序唯一的文档类来进行<sup>[10][11]</sup>。

### 2.2 关键技术介绍

#### 2.2.1 MFC 简介

要开发应用程序，首先必须选择一个编程接口。对于 Windows 平台来说，Windows 的开发者 Microsoft 提供的编程接口产品——MFC (Microsoft Foundation Classes) 库无疑具备其它接口产品所无法比拟的优越性能。

MFC 是一个应用程序框架。也就是说，MFC 是一个源代码或库格式的 C++ 类集。使用这个框架可以在较短时间内创建一个支持高级操作系统功能的完整的应用程序。此外，MFC 还包含了用以构造用户界面的工具。这些工具，特别是图像编辑器、对话框编辑器和资源编译器，使得构造用户界面中的图形元素这项工作变得更加容

易。一个应用程序框架只涉及到应用程序的标准用户界面，而与该应用程序特有的内容无关。换句话说，这个框架能够构造一个功能强大的用户界面，把时间节省下来去编写实现应用程序目的的其它函数。因此不用花很多时间来创建一个和其它一些运行在操作系统环境中具有同样 Windows 化界面的应用程序，因为只需灵活地运用那些库就可以达到这些目的了。

应用程序框架还提供可重用的代码。代码重用是像 C++ 这样的面向对象的编程方法得以发展的主要原因之一。通过将可重用的代码封装成类，可以很容易地使用已经存在的类来添加需要的功能。代码重用使得用户可以使用经过严格测试的代码来构造应用程序，并很容易获得预期的功能，而且还可以很容易地将编写的类与其它需要同样功能的应用程序进行共享<sup>[10][12]</sup>。

## 2.2.2 面向对象 (Object Oriented) 的开发技术

90 年代以后，面向对象的开发技术开始出现在人机界面系统中，这种技术大大提高了人机界面的开发效率，它的出现和发展对软件开发是革命性的。

虽然软件技术在不断地发展，但同时也认识到这种技术的发展始终落后于硬件技术的发展。这种现象在工业自动化领域同样存在，而且这种软件和硬件之间的矛盾显得更加突出。问题的根源在于硬件的开发是一种建立在大量可重用部件，即大规模集成电路芯片的基础之上的。而软件要想得到迅速发展必须也采用同样的或者类似的技术，这就是对象技术 (OOT)。如果在计算机软件中建立了一个“物体”，它具有自己的状态、行为和标识，具有类似的状态和行为的“物体”被划分成为一个类，这些“物体”或者它的实例是可以交换的，可以称之为对象。在利用对象技术的过程中，首先对对象的特征进行抽取即对象的抽象化过程，然后将对象的状态和行为以及它的标识进行封装形成对象的原型；同时在实际应用对象的过程中可以对对象原型进行继承和实例化。这样对象就相当于一个黑匣子，大量的程序代码和数据被封装在这个黑匣子当中，用户只需在对象继承和实例化过程中通过对象的对外接口（公共属性）改变对象的状态和行为以适合自己应用的需要。

利用对象以及对象技术的好处是显而易见的。首先，对象的应用解决了程序设计

过程复杂性的问题,使得问题变得简单化了,再不需要面对大量而复杂的程序代码了,不必彻底了解对象内部的运行机制,只需搞清楚对外接口的定义和编写一些“行为程序”就可以了。其次,当一个特定的对象被封装完成之后,它可以被多次、大量地重复使用,每一次只需对对象进行实例化和继承。这样就大大提高了程序设计的效率,减少了不必要的重复工作。这在某种程度上非常类似于大规模集成电路。

仅就以上的两点优势而言,对象技术非常适合于在人机界面系统中使用。人机界面系统是模块化的软件结构,而且是由大量的、重复使用的目标(区别于对象)组成的,比如,在一个虚拟仪器面板上可能使用数个旋钮来调节不同的系统参数。每一类的目标都具有它们共同的特征,而每一个目标又有属于自己的属性,这非常适合于采用对象的方法进行描述。

面向对象的方法在计算机实时监控系统分析和设计中,因其封装性、继承性、多态性等方面的特性而呈现出广阔的应用前景。用它开发的系统很容易实现二次开发,代码可重用性强,因此具有可靠性高、易维护扩展等特点<sup>[13][14]</sup>。

## 2.3 软件结构设计

根据系统使用要求,其功能模块的划分如下图 2-1 所示:

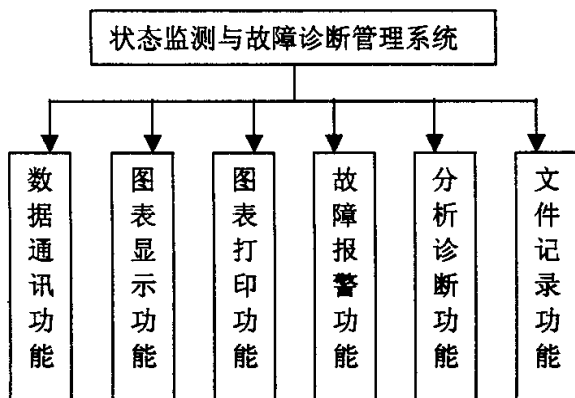


图 2-1 系统软件功能模块划分

### 1) 数据通讯功能模块

包括两个接口的程序设计,一个是串行口连接大字符显示器;另一个是与下位机



的网络接口。串行通信采用 Win32 通信 API 编写，而网络通信则通过 TCP/IP 网络环境下的系统编程界面套接字 (SOCKET) 实现，具体 Winsocket 的编程通过 MFC 提供的 WinSock 类来完成。这部分的功能实现将在第四章中有详细分析，这里不再细说。

## 2) 图表显示功能模块

为了形象显示压力，位移和温度等参数的实时变化，根据系统要求开发了一套虚拟仪表控件，包括棒图控件 C3DBarCtrl，趋势图控件 C3DTrendCtrl，罗盘控件 C3DMeterCtrl，滑块控件 C3DsliderCtrl 等。然后再由这些控件再组成各个虚拟仪表盘。每个虚拟仪表盘视图均派生自 CFormView 类，各视图功能分述如下：

(1) 主界面 CMonitorView 负责以罗盘控件显示主系统、主缸、回程缸、充液罐、左操作机和右操作机回弹压力实际值；以罗盘控件显示充液罐温度变化；以棒图控件显示 1#~4#主泵、主系统、主缸和回程缸压力实际值；以滑块控件显示左、右操作机钳杆侧移实际值。其实际工作画面如图 2-2 所示：

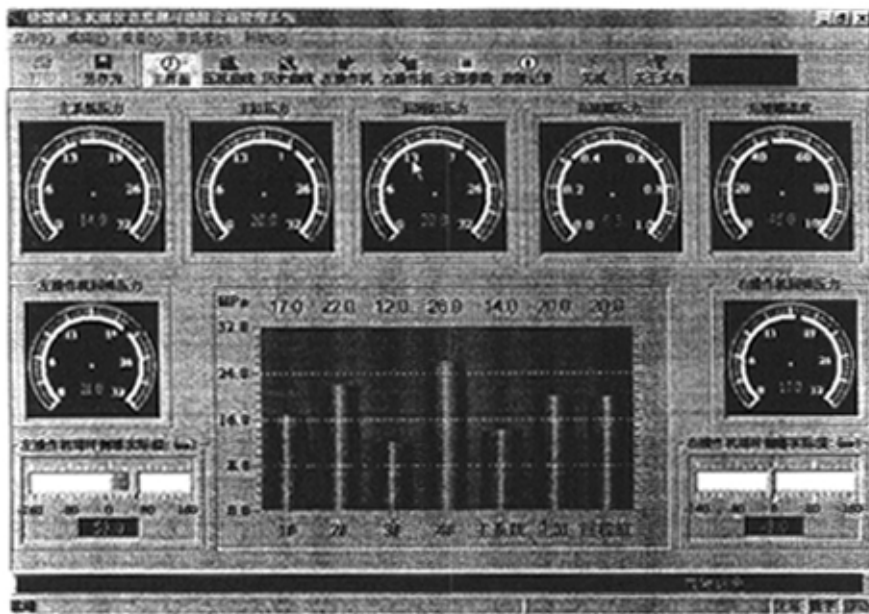


图 2-2 软件系统工作主界面

(2) 压机曲线 CPressRealView 负责以趋势图控件实时动态显示压机实际位移曲线、主系统压力曲线、回程缸压力曲线、主缸压力曲线以及与压机控制有关的模拟量曲线。其实际工作画面如图 2-3 所示：

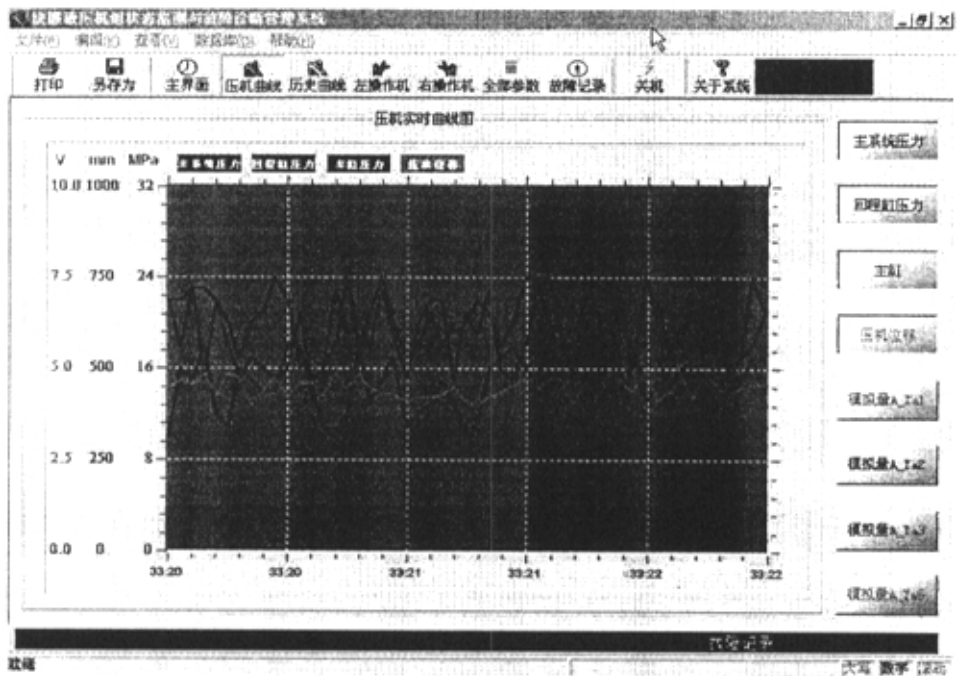


图 2-3 压机实时曲线图

(3) 历史曲线 CPressHistoryView 负责以趋势图控件显示保存在文件中的压机实际位移曲线、主系统压力曲线、回程缸压力曲线、主缸压力曲线以及与压机控制有关的模拟量曲线。其中安排静态控件显示记录保存时间，位图按钮控制记录的打开及翻页。其实际工作画面如图 2-4 所示：

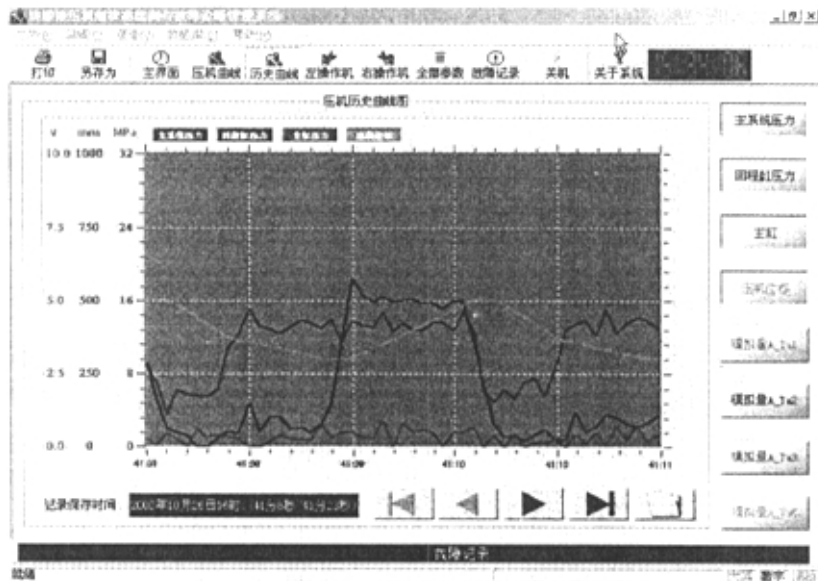


图 2-4 压机历史曲线图

(4) 左操作机 CManipulator1View 负责以趋势图控件实时动态显示左操作机大车行走实际位移曲线、左操作机夹钳旋转实际角度变化曲线、左操作机旋转控制模拟量 C\_Ya9 变化曲线和左操作机行走控制模拟量 C\_Ya10 变化曲线。其实际工作画面如图 2-5 所示：

(5) 右操作机 CManipulator2View 负责以趋势图控件实时动态显示右操作机大车行走实际位移曲线、右操作机夹钳旋转实际角度变化曲线、右操作机旋转控制模拟量 D\_Ya9 变化曲线和右操作机行走控制模拟量 D\_Ya10 变化曲线。其实际工作画面如图 2-6 所示：

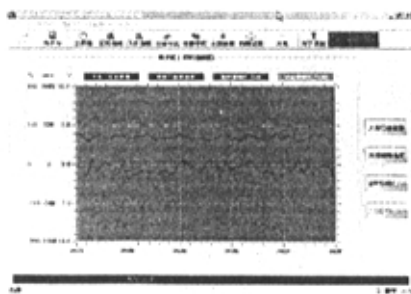


图 2-5 左操作机实时曲线图

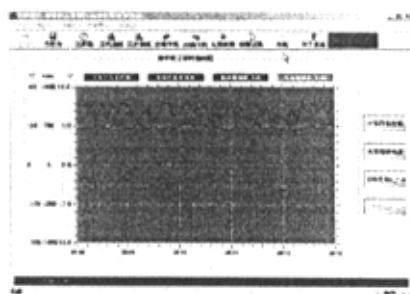


图 2-6 右操作机实时曲线图

# 华中科技大学硕士学位论文

(6) 全部参数 CAHParaView 负责以列表控件实时动态显示全部传送到监测计算机中的参数值。其实际工作画面如图 2-7 所示:

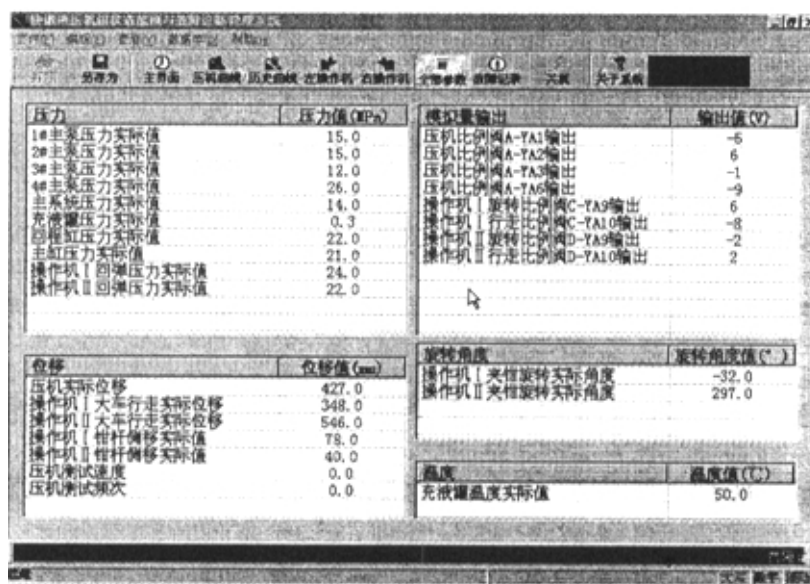


图 2-7 系统全部参数显示

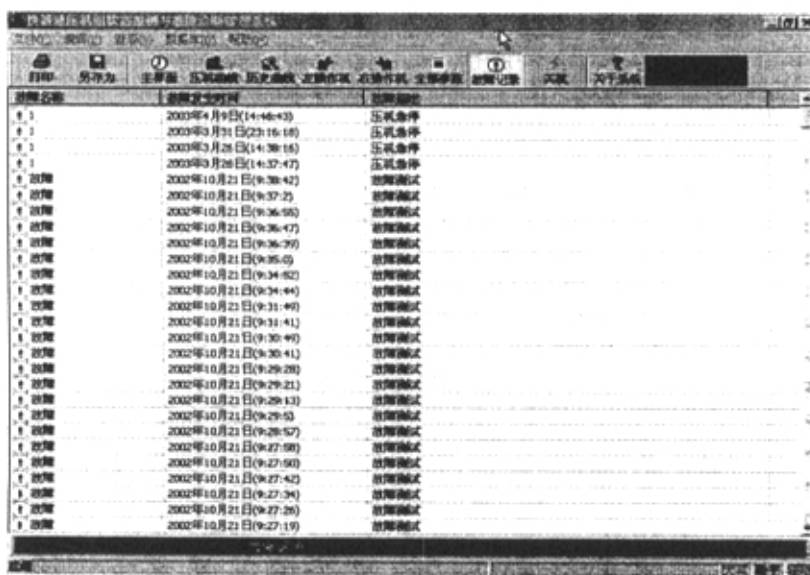


图 2-8 故障记录视图

(7) 故障记录 CFaultView 派生自 CListView 类,它以列表视的形式显示当前及历史上发生的每个故障名称,发生时间及详细描述,并能实时更新。其实际工作画面如图 2-8 所示:

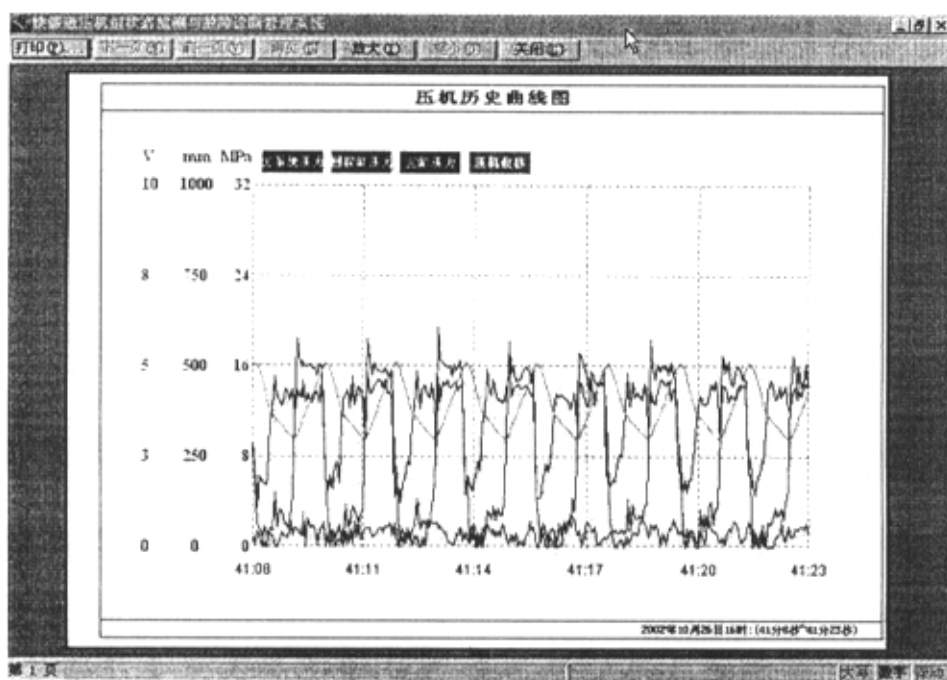
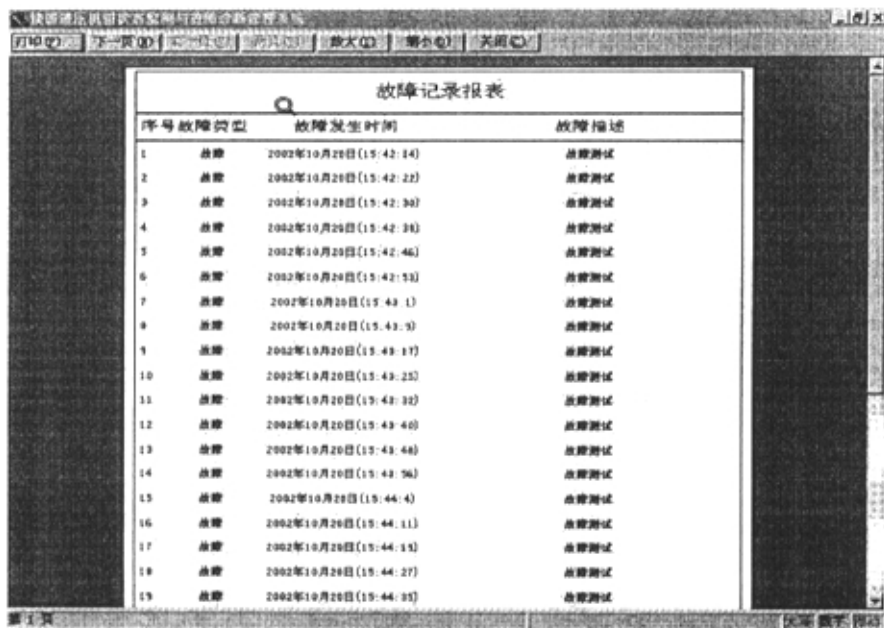


图 2-9 压机历史曲线打印预览

### 3) 图表打印功能

Windows 的 DC 观念,在程序的绘图操作与实际设备的驱动程序之间做了一道隔离,使得绘图操作完全不需修改就可以输出到不同的设备上。MFC 更将整个打印和预览机制都埋在应用程序框架之中了,只需重载 OnPreparePrinting(CPrintInfo\* pInfo) 函数设定打印页数,并在 OnPrint() 函数中编写负责只在打印时才做(在屏幕显示时不做)的操作,例如印出页眉和页脚<sup>[11]</sup>。根据系统要求,压机曲线 CPressRealView 视图、历史曲线 CPressHistoryView 视图和故障记录 CFaultView 列表视需要具备打印及打印预览机制。其实际工作画面如图 2-9 和图 2-10 所示:





序号	故障类型	故障发生时间	故障描述
1	故障	2002年10月20日(15:42:14)	故障测试
2	故障	2002年10月20日(15:42:22)	故障测试
3	故障	2002年10月20日(15:42:30)	故障测试
4	故障	2002年10月20日(15:42:38)	故障测试
5	故障	2002年10月20日(15:42:46)	故障测试
6	故障	2002年10月20日(15:42:53)	故障测试
7	故障	2002年10月20日(15:43:1)	故障测试
8	故障	2002年10月20日(15:43:9)	故障测试
9	故障	2002年10月20日(15:43:17)	故障测试
10	故障	2002年10月20日(15:43:25)	故障测试
11	故障	2002年10月20日(15:43:32)	故障测试
12	故障	2002年10月20日(15:43:40)	故障测试
13	故障	2002年10月20日(15:43:48)	故障测试
14	故障	2002年10月20日(15:43:56)	故障测试
15	故障	2002年10月20日(15:44:4)	故障测试
16	故障	2002年10月20日(15:44:11)	故障测试
17	故障	2002年10月20日(15:44:19)	故障测试
18	故障	2002年10月20日(15:44:27)	故障测试
19	故障	2002年10月20日(15:44:35)	故障测试

图 2-10 故障记录报表打印预览

## 4) 故障报警功能

实时测量数据 `m_DataReal` 中 `InPlcStatus.In` 中记录着液压系统的运行状态，例如压机急停 `Forging_Estop`，操作机急停 `Robot_Estop` 等。通过对这些状态值与故障源之间的一一对应关系，实现故障的诊断及报警，并通过屏幕下方的黑色故障记录状态条 `m_wndFault` 及时显示最新的故障信息。其实际工作画面如图 2-11 所示：



图 2-11 故障记录状态条

## 5) 分析诊断功能

分析诊断功能作为故障诊断的核心功能，目前只处于初步水平。当前的主要工作是将历史故障信息保存在故障数据库中，以方便以后对故障信息的分类整理，进而归纳总结出故障发生的规律性，最终达到利用这些知识指导设备的维修与管理，提高设备的可靠性与使用率的目的。故障数据库设计成 Access 数据库（即\*.MDB 文件），在数据库访问模式上采用在访问 Access 数据库具有很好的性能的 DAO(Database Access Object) 方式，因为 DAO 使用 Microsoft Jet 数据库引擎来访问数据库。Microsoft Jet

为象 Access 和 Visual Basic 这样的产品提供了数据引擎。具体分析放在 5.4 故障数据库设计一节中。以下是数据库在定位（如图 2-11）和记录整理（如图 2-12）上的两个工作画面：

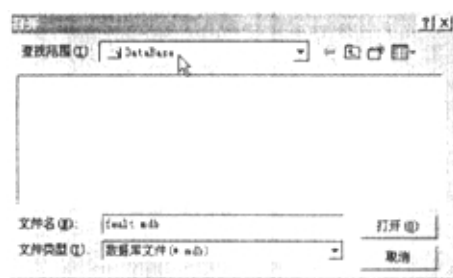


图 2-11 数据库定位对话框

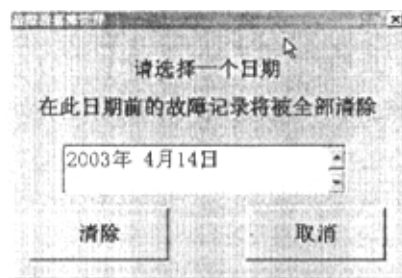


图 2-12 数据库记录管理对话框

## 6) 文件记录功能

负责将任意参数的最近 250 个值记录到文件中，通过 MFC 的 Serialization 机制，重载文档类 CMonitorDoc 中名为 Serialize 的函数。数据的保存和调入采用 MFC 提供的 CFile 类和 CArchive 类，两者配合可方便地将测量数据对象串行化，以流的方式存取。

## 2.4 本章小结

本章全面介绍了快锻液压机组状态监测与故障诊断管理软件系统的总体设计方案，简要介绍了所采用的主要技术手段——MFC 和面向对象软件开发技术。还分析了软件的结构和主要功能模块，并对每个模块要完成的功能作了详细说明。

## 3 计算机监测控制系统研究

在现代工业企业的生产和管理中,大量的物理量、工艺数据、特性参数需要进行实时检测、监督管理和自动控制(简称为监测控制)。这是现代化工业生产必不可少的基本手段。从单台计算机直接监测控制到多级计算机监测控制系统,以及分布式、网络化、智能化的系统,在各种工业企业中都有应用。随着计算机软硬件技术的迅速发展,监测控制系统的应用已发展到相当高的水平。各种类型的商品化硬软件模块和系统,从简单的 I/O 模块到复杂的集散控制系统,在市场上已有供应,在工业生产中也较多的应用。随着现代工业生产向大型、复杂、连续和综合化发展,监测控制的要求越来越高,需要组成更高性能的系统 and 采用更优秀的控制手段<sup>[15]</sup>。

### 3.1 计算机监测控制系统

#### 3.1.1 监测控制计算机

从计算机应用角度出发,可将计算机分为面事务处理的计算机和面向实时过程的计算机。面向事务处理的计算机大多采用通用计算机,包括通用大型机、小型机和微型机(个人计算机)。面向实时过程的计算机也称过程计算机或监测控制计算机(简称监控计算机),它除了具有通用计算机的一些共同特性外,在系统结构、硬软件组成、设计思想、开发工具、使用方法等方面与通用计算机有许多不同<sup>[16]</sup>。其主要差别可归纳为:

- 严格的可靠性措施;
- 丰富的过程输入输出(I/O)接口;
- 多样的人机接口;
- 实时操作系统或保证实时性的软件技术。

监测控制计算机所含的硬软件根据实际应用的需要而不相同,图 3-1 给出了一种典型的分类,一个具体的监测控制计算机可以包括其中的某些部分。在实际应用中,监测控制计算机的种类很多。在我国,应用得较多的有下述的几种类型:

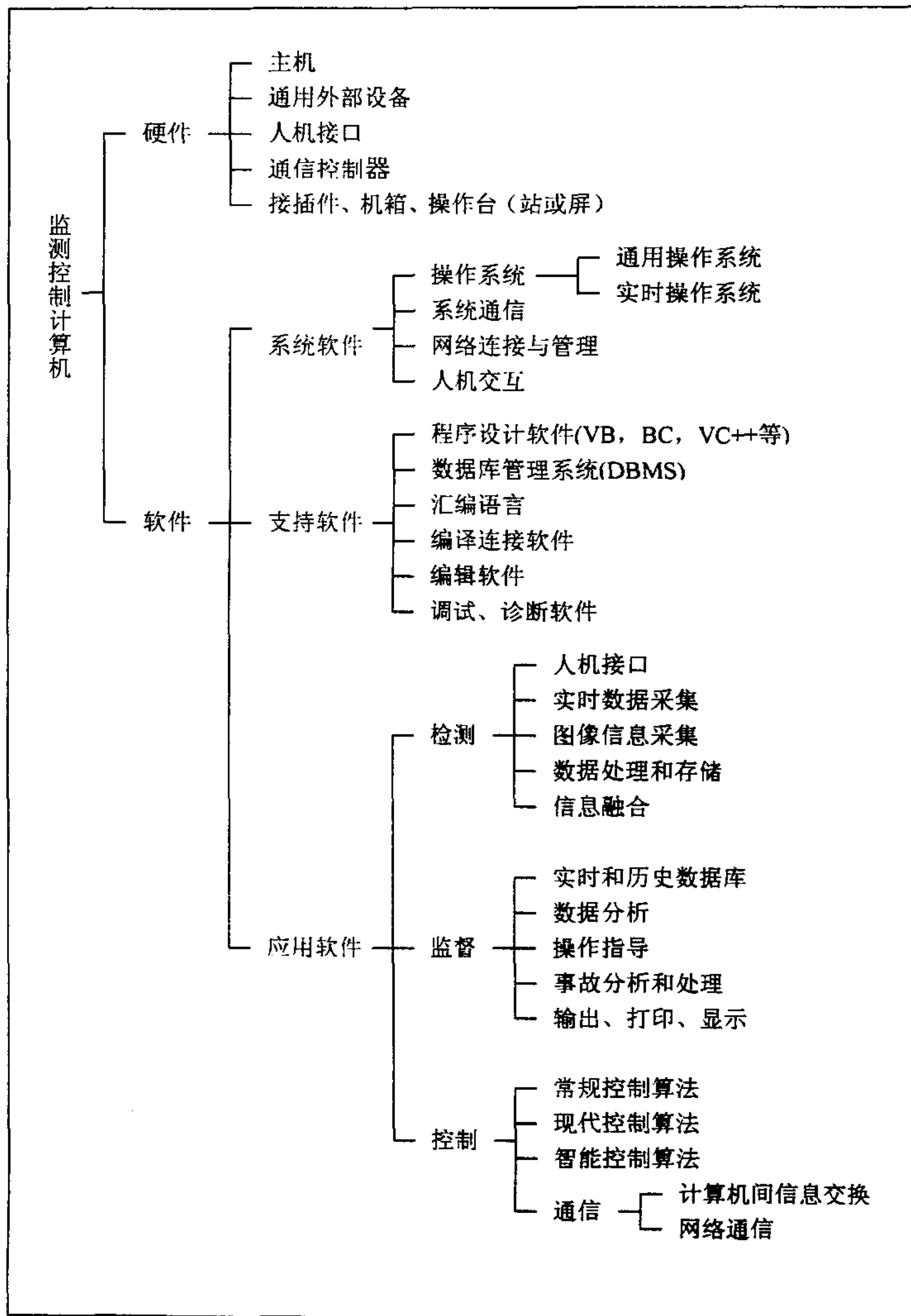


图 3-1 监测控制计算机的组成

1) 数字控制器 主要分为两类：一类是专用的小型控制器，有一定的应用范围，通用性不强。一般是针对某些或某类应用系统专门设计制造的，针对性强，性能专一，操作简单，方便实用；另一类是可编程序控制器（programmable controller），也称可编程序逻辑控制器 PLC（programmable logic controller）。现代 PLC 充分应用了计算机技术，除具有传统的逻辑、计时、计数等功能外，还增加了运算、数据传送和处理、通信等功能，大大提高了其性能表现及应用范围。

2) 工业控制计算机 工业控制计算机产品按总线类型可划分为 VME 总线工控机、MULTIBUS 总线工控机、STD 总线工控机、PC 总线工控机等。其中发展最快的是 PC 总线工控机（industrial personal computer, IPC），由于 IPC 有严格的标准，有丰富的外围接口板，有的还配有功能颇强的组态软件或工业控制软件包，给用户提供了良好的开发和应用条件，因而已成为当前工控机的主流。

3) 嵌入式 PC 机 这是一种近年来发展起来的超小型化的 PC 机，外形尺寸为 95mm×90mm，功耗小，无需机箱和底板就可以将有关的模块直接叠装组合成各种系统。目前嵌入式 PC 正处在迅速发展之中，产品种类很多，规格型号不统一。有代表性的产品是 PC/104。

### 3.1.2 计算机监测控制系统

本文中所讨论的监测控制系统（简称计算机监控系统），是以监测控制计算机为主体，加上检测装置、执行机构、与被监测控制的对象（生产过程）共同构成的整体，图 3-2 示出了一个典型的计算机监测控制系统。在这个系统中，计算机直接参与生产过程的检测（monitor）、监督（supervise）和控制（control），或者说应具有下述三方面的功能<sup>[17]</sup>：

1) 采集与处理功能 主要是对生产过程的参数进行检测、采样和必要的预处理，并以一定的形式输出（如 CRT 屏幕显示和打印制表），为生产人员提供详实的数据，以便于他们分析、了解生产情况，监视生产过程的进行。

2) 监督功能 将检测到的实时数据、人工输入的数据等信息进行分析、归纳、整理、计算等二次加工，并制成实时和历史数据加以存储。根据实际生产过程的需要



及生产进程的情况，进行工况分析、故障诊断、险情预测，并以图、文、声等多种形式及时须做出报道，以进行操作指导、事故报警。监督系统的输出一般都不直接作用于生产过程，而是经过生产人员的判断后再由操作人员对生产过程进行干预。

3) 控制功能 在检测的基础上进行信息加工，根据事先决定的控制策略形成控制输出，直接作用于生产过程。

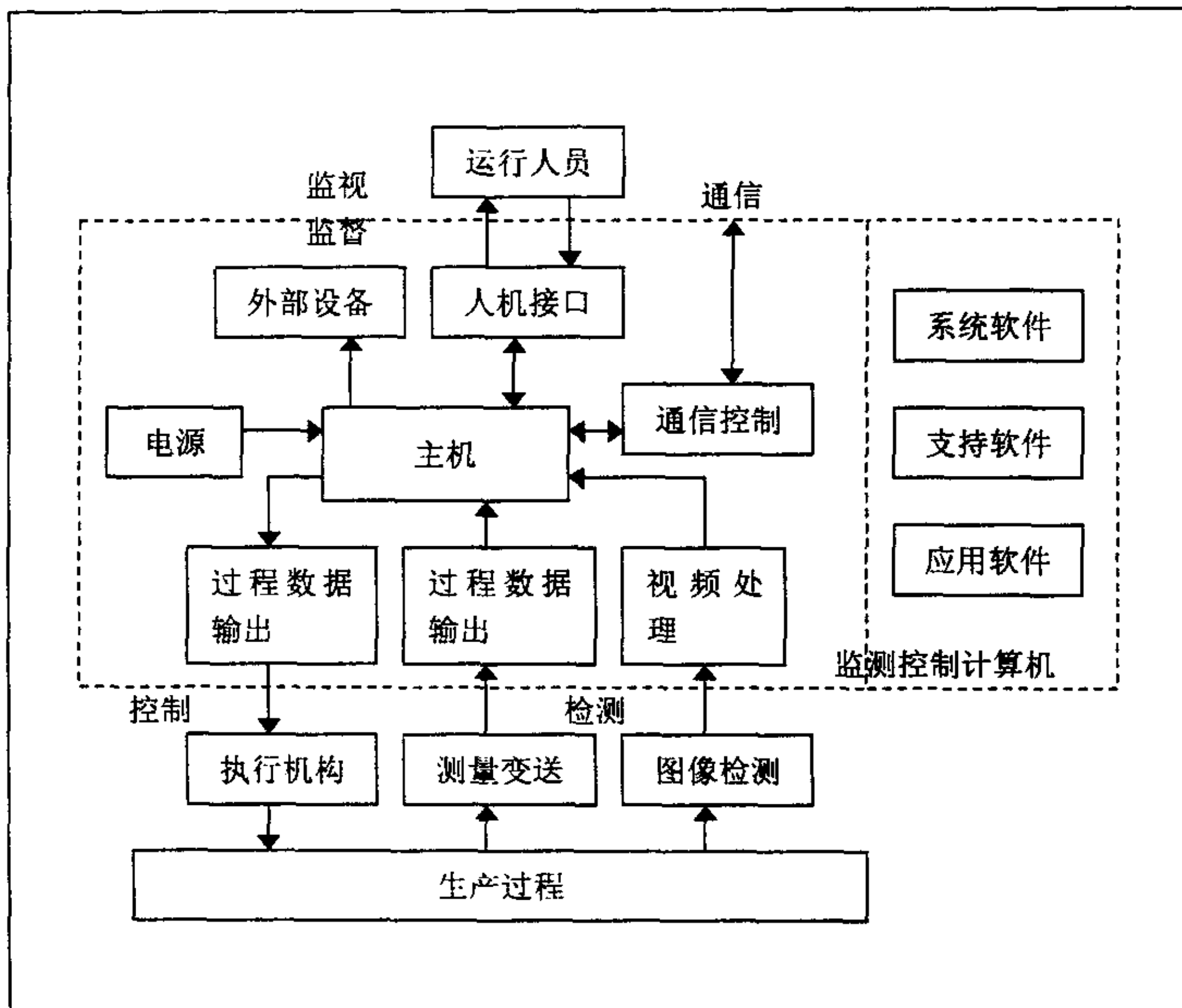


图 3-2 计算机监测控制系统示意图

完整的计算机监测控制系统是上述三种功能的综合集成，它利用计算机高速度、大容量和智能化的特点，可以把一个复杂的生产过程组织管理成为一个综合、完整、高效的自动化整体。当然，在实际使用中，可以根据实际对象的需求情况，系统只具有上述一项或两项功能：或是以某一项为主，而辅以其他的功能。这样可以更针对实际应用的需要，以降低成本，减少复杂性，增强可维护性。

## 3.2 计算机监测控制系统的特点

### 1) 实时性

计算机监测控制系统是一种实时计算机系统，实时性是它区别于普通（通用）计算机系统的键特点，也是衡量计算机监测控制系统性能的一个重要指标。实时性是系统对外界激励（事件）及时作出响应的能力，即系统能在多少时间内响应外部事件的发生。这一涵义常用“系统响应时间”来衡量。计算机监测控制系统工作的正确性不仅依赖于它计算、推理结果的逻辑正确性，而且还依赖于得出结果的时间<sup>[18]</sup>。

### 2) 可靠性

计算机监测控制系统的可靠性是指系统无故障运行的能力。可靠性常用“平均无故障运行时间”，即平均的故障间隔时间 MTBF（mean time between failures）来定量的衡量<sup>[19]</sup>。我国《微型通用计算机技术条例》（国标 GB9813-88）规定，一般通用微机的 MTBF 不得低于 2000 小时，板级产品的 MTBF 不得低于 4000 小时。工业控制计算机的 MTBF 就要大得多，例如，Prolog 公司的 STD System II 的 MTBF 大于 12 年，Intel 公司的工业 PC 机主机系统的 MTBF 是  $1.6 \times 10^5$  小时，即大于 18 年<sup>[20]</sup>。

### 3) 可维护性

可维护性是指进行维护工作时的方便快捷程度。计算机监测控制系统的故障会影响工业生产过程的正常操作，有时会大面积地影响生产过程的进行，甚至使整个生产瘫痪。因此，方便地维护监测控制系统的正常运行，在最短时间内排除它的故障成为计算机监测控制系统的一个重要特点。

### 4) 过程量采集及输出

监测控制系统的一个突出的特点是具有强大的 I/O 功能，即大量的现场信息要直接从工业现场采集并送入计算机中。从当前已有的应用来看，有两大信息：

(1) 数据信息 主要有三种类型的数据输入信号：模拟量输入（AI）、开关量输入（SI）、脉冲量（PI）。对应于三种类型的数据输入信号，有三种类型的数据输出信号：模拟量输出（AO）、开关量输出（SO）、脉冲量输出（PO）。它们用来控制执行机构，或送到有关的显示、报警、记录设备中。

(2) 图像信息 常用的获取图像信息的设备是工业摄像机，它摄取的视频图像通过视频处理卡后直接进行计算机。在常规的使用中，视频图像是直接输出到屏幕上作显示的。最近的技术发展中，视频图像经图像处理后再进行显示；或再与数据信息融合后作为控制信号使用。在本课题研制的系统中就包括对工业摄像机的常规应用，用于监视生产现场。

## 5) 人机交互

在计算机监测控制系统中，人机交互的方式比较丰富。特别是在复杂、大型、综合、连续的生产过程中，操作人员要在短时间内接受多个信息，进行分析判断，完成有关操作，因此要求监测控制系统具有多种而不是单一的人机交互方式。除常规的键盘、鼠标、CRT 显示器外，通常还有触摸屏、专用键盘、大屏幕显示、语音等。本课题所开发的系统中就使用了触摸屏，大字符显示器等特殊设备，以满足生产过程中丰富的人机交互要求。

## 3.3 快锻液压机组计算机控制系统的设计

快锻液压机组计算机控制系统由 PLC 网络，工业控制和监测计算机组成，其组成框图如图 3-3 所示。

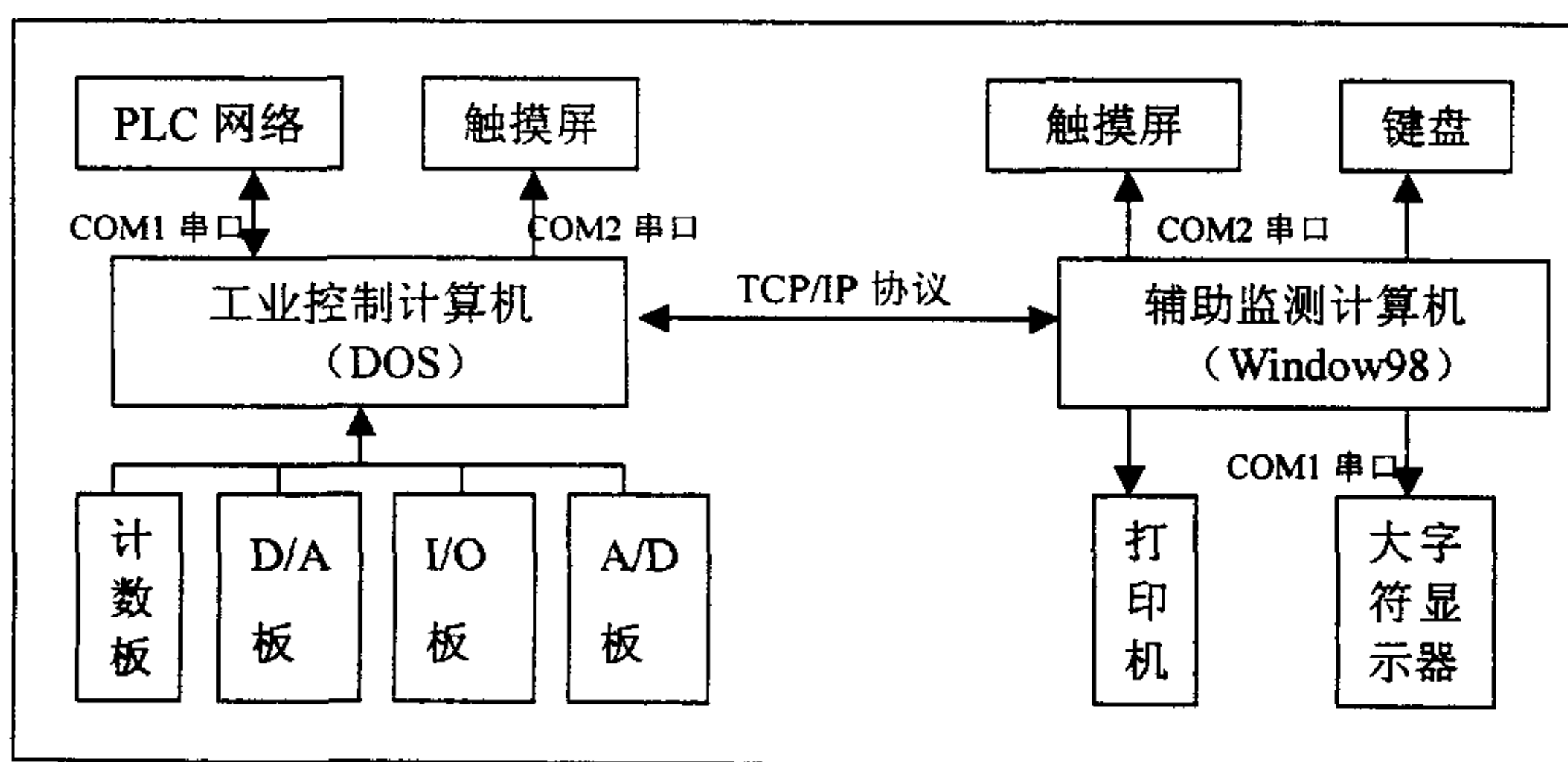


图 3-3 快锻液压机组计算机控制系统组成框图

整个系统采用主从机结构，下位机（工业控制机）完成半自动，自动及联动控制

以及系统的压力、位移、温度等数据的实时采集；上位机（监测计算机）动态显示，记录和打印系统的监测参数和状态，完成信号的分析处理与故障诊断。工控机通过串行口与 PLC 网络进行通信，从 PLC 中获取控制台开关及按键的状态，同时向 PLC 写入有关的控制参数。上位机与下位机之间通过网卡相连，运行 TCP/IP 协议进行通信。PLC 部分完成系统的手动控制。

在硬件组成上，系统中上、下位机采用的是工控机，属于监测控制计算机分类中的第二类——PC 总线工控机，是当前监测控制系统开发中的主流硬件配置。该机型的优点是工作稳定，并且配有丰富的外围接口板。另外，由于使用了新的人机交互设备——触摸屏，整个系统的可操作性和人机交互能力大大增强，便于操作人员的使用维护和推广应用。

在系统控制上，一个典型的计算机监测控制系统所应具备的三种基本功能被合理的分配到系统的各部分中。下位机与 PLC 网络相连，通过与 PLC 网络的通信间接实现控制功能，而由 PLC 网络直接控制生产设备及过程。同时下位机还要实现数据采集与处理功能。而监督功能则由上位机实现。

在系统软件方面，负责信号采集的下位机采用实时性强的 DOS6.22 操作系统，满足系统需要进行大量数据实时采集的需要；负责在线监测与故障诊断的上位机采用 Windows98 操作系统，以充分发挥 Windows 图形化界面，多任务处理，动态数据共享等特性完成复杂的数据分析与处理。

而应用软件包括两部分，一部分运行于下位机上，功能包括：与 PLC 网络之间的串口通信及控制参数的传输；与上位机之间的网络通信；快锻液压机组的压力、温度、位移等信号的实时采集与预处理。另一部分就是本课题所研究开发的快锻液压机组状态监测与故障诊断管理系统。该系统运行在上位机（监测计算机）上。功能包括：动态显示，记录和打印系统的监测参数和状态，完成信号的分析处理与故障诊断。

### 3.4 本章小结

本章对计算机监控系统进行了研究，分析了其软硬件组成及性能特点。最后结合课题完成了快锻液压机组计算机控制系统的设计。

## 4 数据通讯接口设计

数据通信功能模块是系统中很重要的一部分，由于采用主从机结构，应用程序的很多参数及状态信号需要在不同计算机之间进行传输，因此网络通信是必不可少的条件。而要解决通信问题，关键就是完成一个好的通讯接口设计。本系统是面向 PC 机平台的应用，并且决定采用客户 / 服务器结构来架构应用程序，由于 TCP/IP 协议应用的广泛性、完善性和跨平台性，因此采用基于 TCP / IP 协议的 Winsock 接口实现上位机和下位机间的数据传输。另外，与监测计算机相连的还包括一个大字符显示器，所以应用程序中还包括一个串口通讯模块的设计。下面分别就串行通信接口程序的开发及网络通信程序的开发详细讨论其原理及实现。

### 4.1 串行通信接口程序的开发

本系统中的串行通信接口程序是为了解决上位机（监测计算机）与大字符显示器之间的数据传输而开发的。由于大字符显示器本身不需要编程驱动，所以只需按照大字符显示器规定的输入输出接口与数据格式进行数据传输，就可以使大字符显示器正常工作。该型大字符显示器采用串行口作为接口，其正常工作的配置为：

- 波特率 9600bps;
- 无校验;
- 8 位数据位;
- 2 位停止位。

正常工作时它通过内部逻辑驱动红色的 LED 灯显示所组成的四个数字，范围从 0~9999。其值是通过串口接受两个字节数据来确定的。至于大字符显示器是如何把接受到的值转化成 LED 灯的显示组合，并不需要我们去考虑。需要注意的是这两个字节的数据不能同时发送，中间必须留有一定的时间间隔。目的是保证两个字节能先后被接受，前一个接受的字节用来确定前两个数字，后一个接受的字节用来确定后两个数字。



下面就详细的介绍一下串行通信的原理,实现方式和本系统中串行通信功能的具体实现步骤。

## 4.1.1 串口通信编程概述

经过多年的使用和发展,目前已经有多种串行通信接口标准。但都是在 RS-232 标准的基础上经过改进而形成的。RS-232C 标准是美国 EIA (电子工业联合会) 与 BELL 等公司一起开发并于 1969 年公布的通信协议。它适合于数据传输速率在 0~20000bit/s 范围内的通信。由于通信设备厂商都生产与 RS-232C 制式兼容的通信设备,因此,它作为一种标准,目前已在微机通信接口中广泛采用,它不仅已被内置于每一台计算机,同时也已被内置于从微控制器到主机的多种类型的计算机及其相连接的设备。

RS-232 有几个优点:

- RS-232 应用广泛。每一台 PC 都有一个或多个 RS-232 端口。
- 在微控制器中,接口芯片使得一个 5V 串口转换成 RS-232 变得非常容易。
- 连接距离可以达到 50~100 英尺。
- 对于双向连接,只需要 3 条导线。

在 DOS 系统接口中 DOS INT21H 的 03H 和 04H 号功能调用为异步串行通信的接收和发送功能,而 BIOS INT14H 有 4 组功能调用为串行通信服务,正因为如此,在 DOS 中采用寄存器直接读写、BIOS 调用、通信中断程序等方法可以比较容易实现串口通信。

Windows 环境下地串口编程与 DOS 环境下的串口编程有很大不同。由于 Windows 采用消息驱动和设备统一管理,以及利用消息队列进行程序控制,所以 DOS 下的方法都不宜采用。Windows 环境下的编程的最大特征之一就是设备无关性,它通过设备驱动程序将 Windows 应用程序同不同的外部设备隔离。Windows 封装了 Windows 的通信机制,这种方式称为通信 API,Windows 程序员可以利用 Windows 通信 API 进行编程,不用对硬件直接进行操作。这种体制被称为 Windows 开放式服务体系 (WOSA, Windows Open Services Architectures)。

早期的 Windows 3.x 与 Windows 9x/NT/2000 的通信 API 也有很大的不同, 在 16 位的串行通信程序中, 一般使用 16 位的 Windows API 通信函数。16 位下的串口通信程序最大的特点就在于串口等外部设备的操作有自己特有的 API 函数, 例如 `OpenComm()`, `CloseComm()`, `ReadComm()`, `WriteComm()` 等。而在 32 位的 Windows 系统中, 串口和其他的通信设备是作为文件处理的。串口的打开、关闭、读取和写入所用的函数与操作文件的函数完全一致。Windows 9x/NT/2000 中的 API 一般都支持 32 位的操作, 因此又称为 Win32 API。为了在上述系统中实现串行数据传送, 可以使用 Win32 通信 API<sup>[21][22]</sup>。

## 4.1.2 Win32 中实现串口通信的方法

串行端口的本质功能是作为 CPU 和串行设备间的编码转换器。当数据从 CPU 经过串行端口发送出去时, 字节数据转换为串行的位。在接收数据时, 串行的位被转换为字节数据。在 Windows 环境 (Windows NT、Win98、Windows2000) 下, 串口是系统资源的一部分。应用程序要使用串口进行通信, 必须在使用之前向操作系统提出资源申请要求 (打开串口), 通信完成后必须释放资源 (关闭串口)。由于本系统的工作环境在 Win32 平台上, 下面就重点讨论一下在 32 位模式下实现串行通信有那几种方法<sup>[21]</sup>:

- 以文件方式打开串口

Windows 的 SDK 提供了完备的 Win32 API 函数, 使编程较为容易。

- 使用现有的 ActiveX 控件实现

现在一般使用的是微软公司的 MSComm 控件, 它是作为一个 ocx 来提供的。基本的使用流程和直接使用 API 函数类似。

- 直接嵌入式汇编法

这种方法不能在 Windows NT 操作系统下使用, 由于 Windows98 的非完全保护, 可以在 Windows98 系统下使用这种操作方式。利用 C/C++ 汇编嵌入功能, 可以直接对串行口进行操作。

- 编写 VXD 或者 WDM 实现

可以参考 Walter Oney 的《System Programming for Windows 95》以及《Programming the Microsoft Windows Driver Model》。

下面结合系统要求具体分析采取各种方法的优劣。直接嵌入式汇编法和编写 VXD 或者 WDM 实现这两种方法主要目的是为了提高串口操作和速度，后者还能满足用户对串口通信的特殊要求，由于本系统中与大字符显示器之间传输的不是实时性要求很高的控制数据，只是为了达到将某一物理量放大显示，以便工作人员能更好地了解当前设备的工作状态。而且这两种方法编程复杂，可移植性差，用在这里不太合适。而在 Visual C++6.0 开发环境中使用 ActiveX 控件虽然也很方便，但缺乏必要的灵活性，只有以文件打开串口这种方法用在本系统中最为适合。使用这种方式不仅能够灵活的操作串口，而且编程也比较容易。下面就简单介绍在 32 位 Windows 环境下以文件方式对串口进行操作的流程，可以分成如下步骤：

1) 应用 CreateFile ( ) 来打开串口

如果调用成功，那么该函数返回文件的句柄，如果调用失败，则函数返回 INVALID\_HANDLE\_VALUE。

例如下面所示的代码就是串行通信接口程序中用来打开 COM1 端口的：

```
HANDLE hCom;
DWORD dwError;
hCom=CreateFile ( "COM1" ,//文件名
GENERIC_READ|GENERIC_WRITE,//允许读和写
0, //独占方式
NULL,
OPEN_EXISTING,//打开而不是创建
FILE_ATTRIBUTE_NORMAL|FILE_FLAG_OVERLAPPED,//重叠方式
NULL
);

if (hCom==INVALID_HANDLE_VALUE)
```

```
{  
    dwError=GetLastError();  
    ...//处理错误  
}
```

## 2) 串行口的初始化

对串行口进行初始化工作需要通过一个 DCB 结构来进行。DCB 结构包含了诸如数据传输速率、每个字符的数据位数、奇偶校验和停止位数等信息。在查询或配置串行口的属性时，都需要用 DCB 结构来作为缓冲区。

调用 GetCommState 函数可以获得串口的配置，该函数把当前配置填充到一个 DCB 结构中。一般在用 CreateFile 打开串行口后，可以调用 GetCommState 函数来获取串行口的初始配置。要修改串行口的配置，应该先修改 DCB 结构，然后再调用 SetCommState 函数用指定的 DCB 结构来设置串行口。

本程序根据大字符显示器串行接口的正常工作配置将对应 DCB 数据结构中参数初始化为：dcb.BaudRate=CBR\_9600；dcb.Parity=NOPARITY；dcb.ByteSize=8；dcb.StopBits=TWOSTOPBITS。

3) 利用 ReadFile() 和 WriteFile() 对串行端口进行读写操作，完成数据传输过程。读写串口 API 是两个很复杂的函数，并且它们根据 CreateFile() 调用中是否设置了 FILE\_FLAG\_OVERLAPPED（重叠方式）有着很不相同的调用表现。下面将会用单独一个小节来讨论读写串口需要注意的问题，同时说明它在本接口程序中的具体应用。

4) 串口使用完后，不能忘了调用 CloseHandle() 关闭串口。

### 4.1.3 读写串口需要注意的问题

利用 Win32 通信 API 读写串口时，可以采用阻塞和非阻塞两种方式，也即同步执行方式和重叠（异步）执行方式。在同步执行时，函数直到读或写操作全部完成后才能返回。由于操作可能需要任意长的时间才能完成，这意味着在同步执行时线程会

被阻塞,从而导致效率降低。在重叠执行时,一旦调用 ReadFile()、WriteFile(),就能立即返回,即使操作还未完成。费时的 I/O 操作在留在后台进行,这样线程就可以做其他的工作。例如,线程可以在不同的句柄上同时执行 I/O 操作,甚至可以在同一句柄上同时进行读写操作。“重叠”一词的含义就在于此。

非阻塞通信中要使用 OVERLAPPED 结构。首先 CreateFile() 函数的 dwFlagsAndAttributes 参数需设为 FILE\_FLAG\_OVERLAPPED 标识;然后在 ReadFile() 时 lpOverlapped 参数一定不能为 NULL,必需指定 OVERLAPPED 结构;接着检查函数调用的返回值,调用 GetLastError(),看是否返回 ERROR\_IO\_PENDING;如是,最后调用 GetOverlappedResult() 返回重叠操作 (overlapped operation) 的结果。如果该函数调用成功,则返回值不为零,如果函数调用失败,则返回值为零,如果想获得进一步的消息,可以调用 GetLastError() 函数来获取。写串口函数 WriteFile() 的使用类似<sup>[23]</sup>。

在设置了异步 I/O 操作后, I/O 操作和函数返回有以下两种情况:

1) 函数返回时 I/O 操作已经完成:此时结果好象是同步执行的,但实际上这是异步执行结果。

2) 函数返回时 I/O 操作还没有完成:此时一方面,函数返回值为零,并且 GetLastError() 函数返回 ERROR\_IO\_PENDING;另一方面,系统把 OVERLAPPED 中的信号事件设为无信号状态,当 I/O 操作完成时,系统要把它设置为信号状态。

通过以上的介绍,下面结合课题来具体分析是如何实现异步写操作的。由于大字符显示器只是被动接受数据,所以监测计算机与大字符显示器之间的串口通信只是单向的,即只有写串口操作。下面的代码是程序自定义函数 WriteCommBlock (unsigned char \* lpBlock, DWORD dwBytesToWrite) 中的主要部分。每调用该函数一次就会对串口写入一字节的数据。

```
CEvent event (FALSE,TRUE);  
OVERLAPPED overlapped;  
// 初始化 OVERLAPPED 结构  
memset (&overlapped, '\0', sizeof (OVERLAPPED));
```

```
overlapped.hEvent = event;
if (!m_hCom) return (FALSE);
fWriteStat=WriteFile (m_hCom,lpBlock,dwBytesToWrite,
    &dwBytesWritten,&overlapped); //执行异步写操作, 函数立即返回
if (!fWriteStat) {
    if (GetLastError () ==ERROR_IO_PENDING) {
        WaitForSingleObject (event,INFINITE); // 等待事件状态信号
        while (!GetOverlappedResult (m_hCom, //返回重叠操作结果
            &overlapped,&dwBytesWritten,TRUE))
        {
            dwError=GetLastError ();
            if (dwError == ERROR_IO_INCOMPLETE)
                continue;
            else {
                ClearCommError (m_hCom,&dwErrorFlags,&ComStat);
                break;
            }
        }
    }
    else {
        ClearCommError (m_hCom,&dwErrorFlags,&ComStat);
        return ( FALSE );
    }
}
```

在程序中使用了等待函数 WaitForSingleObject () 来检查事件对象的当前状态或等待 Windows 状态信号, 下面作一简要说明, 其函数声明如下:

DWORD WaitForSingleObject (

```
HANDLE hHandle,  
DWORD dwMilliseconds  
);
```

该函数如果调用成功，则返回值标识导致该函数返回的事件，否则返回值为 WAIT\_FAILED。如果想获得进一步的错误信息，可以调用 GetLastError() 函数来获取<sup>[24]</sup>。成功的返回值可以是以下的三个值之一：

WAIT\_ABANDONED：等待的对象是互斥对象，并且拥有该对象的线程在终止时候没有释放该对象；

WAIT\_OBJECT\_0：指定的对象处于有信号状态；

WAIT\_TIMEOUT：超时事件发生后，指定的对象仍处于无信号状态。

## 4.2 网络通信程序的开发

本系统中由于 COM1 和 COM2 口分别用于触摸屏和大字符显示器，所以（上位机）监测计算机和（下位机）控制计算机之间的通信采用网卡连接。通讯协议则选择 TCP/IP 协议，因为 TCP/IP 协议是一种比较完善的网络协议，并已成为异种网络和计算机间互联的工业标准。由于下位机在 DOS 环境下工作，所以须加载 NWTCP 网络驱动程序与 Win98 环境下工作的上位机组成 TCP/IP 的网络环境，从而实现跨平台的计算机互联。TCP/IP 网络环境下的应用程序通过系统编程界面套接字（SOCKET）实现。

### 4.2.1 通信协议简介

通信协议用于不同的网络设备之间的信息交换，它们建立了设备之间互相识别的宝贵信息机制。

#### 1) TCP/IP 协议

在众多协议中，传输控制协议/网际协议（TCP/IP）是业界标准的协议组，它是为了跨越局域网和广域网环境的大规模网络而设计的。

TCP/IP 始于 1969 年，也就是美国国防部委任高级资源计划机构网络（ARPANET）



的时候。ARPANET 是资源共享实验的结果。其目的是在美国不同地区和各种超级计算机之间提供高速网络通信链路。

Telnet（用于虚拟终端仿真）和 FTP（文件传输协议）之类的早期协议最早开发是规定经过 ARPANET 共享信息所需的基本实用程序。随着 ARPANET 在规模和作用范围的日益扩大，由此出现了其他两个重要协议：

- 1974 年，传输控制协议（TCP）作为规范草案引入，描述了如何在网络上建立可靠的、主机对主机的数据传输服务。
- 1981 年，网际协议（IP）以草案形式引入，描述了如何在互连网络之间实现寻址的标准以及如何进行数据包的路由。

1983 年 1 月 1 日，ARPANET 开始要求所有的网络通信和基本通信都使用 TCP 和 IP 协议标准。从那天开始，ARPANET 逐渐变成当今著名的 Internet，它所要求的协议逐渐变成 TCP/IP 协议组。

TCP/IP 协议组在各种 TCP/IP 软件中实现，可用于多种计算机平台。今天，TCP/IP 在 Internet 上广泛使用，并经常用于建立大的路由专用互连网络<sup>[25]</sup>。

## 2) TCP/IP 模型

传统的国际标准化组织（ISO）提议的“开放系统互连（OSI）”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、网路层、传输层、会话层、表示层和应用层。

TCP/IP 协议并不完全符合 OSI 的七层参考模型，它采用了 4 层的层级结构，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：应用程序间沟通的层，如简单电子邮件传输（SMTP）、文件传输协议（FTP）、网络远程访问协议（Telnet）等。
- ◇ 传输层：在此层中，它提供了节点间的数据传送服务，如传输控制协议（TCP）、用户数据报协议（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。
- ◇ 互连网络层：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）。

- ◇ 网络接口层：对实际的网络媒体的管理，定义如何使用实际网络（如 Ethernet、Serial Line 等）来传送数据。

如下图 4-1 所示，TCP/IP 模型的每一层都对应于的七层“开放系统互连（OSI）”参考模型中的一层或多层<sup>[25][26]</sup>。

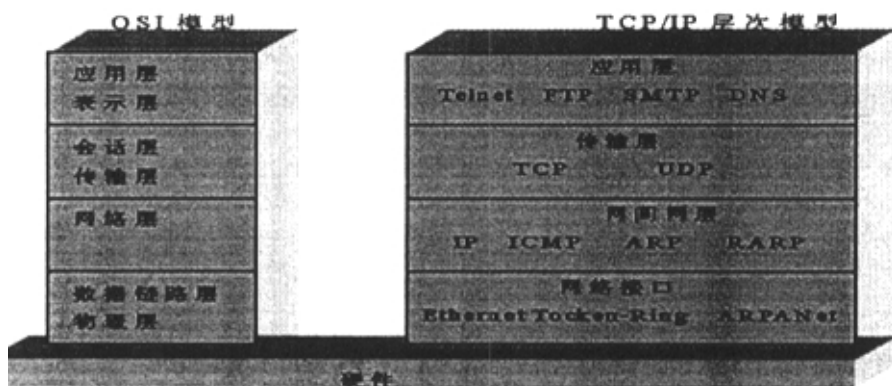


图 4-1 TCP/IP 模型与 OSI 参考模型的比较

### 3) Windows 的 TCP/IP

Windows TCP/IP 是：

- 基于工业标准网络协议的网络软件。
- 支持 Windows 计算机与局域网和广域网环境进行连接的可选择路由的企业网络协议。
- 用于 Windows 计算机和不同系统进行连接并共享信息的核心技术和实用程序。
- 访问全局 Internet 服务，如 World Wide Web 和文件传输协议（FTP）服务器的基础。
- 强健的、可缩放的、跨平台的客户/服务器框架。

因为多数现代操作系统都支持 TCP/IP 协议组，所以可以使用 TCP/IP 与多种系统和资源共享信息。

### 4) 了解 TCP/IP 核心协议

## (1) 网际协议 (IP)

IP 是所需的 TCP/IP 标准, 在 RFC791 “网际协议” 中定义。IP 是无连接的、不可靠的数据报协议, 主要负责在主机之间寻址和选择数据包的路由。

无连接意味着交换数据之前不能建立会话。不可靠意味着传递没有担保。IP 总是尽力传递数据包。IP 数据包可能丢失, 不按顺序传递, 重复或延迟。IP 不尝试从这些错误类型中恢复。所传递的数据包确认以及丢失数据包的恢复是更高层协议的责任, 如 TCP。

## (2) 用户数据报 (UDP)

用户数据报协议 (UDP) 是 TCP/IP 标准, 在 RFC768 “用户数据报协议 (UDP)” 中定义。某些程序使用 UDP 来代替 TCP, 在 TCP/IP 主机之间快速、轻便、不可靠地传输数据。

UDP 提供尽量传递的无连接数据报服务, 这意味着 UDP 无法保证任何数据报的传递或验证数据报的顺序。需要可靠通信的源主机必须使用 TCP 或提供自身顺序和确认服务的程序。

## (3) 传输控制协议 (TCP)

传输控制协议 (TCP) 是所需的 TCP/IP 标准, 在 RFC793 “传输控制协议 (TCP)” 中定义, 并提供可靠的、面向连接的数据报传递服务。TCP 基于两个网络主机之间的点到点通信。

两个 TCP 主机在可以交换数据之前, 必须先相互建立对话。TCP 会话通过三路握手的过程初始化。这个过程使序号同步, 并提供在两个主机之间建立虚拟连接所需的控制信息。一旦初始的三路握手完成, 在发送和接收主机之间按顺序发送和确认段。关闭连接之前, TCP 使用类似的握手过程来验证两个主机都完成发送和接收全部数据 [25] [26] [27]。

### 4.2.2 Windows Sockets 简介

#### 1) 背景

20 世纪 80 年代初, 美国政府的高级研究工程机构 (ARPA) 给加利福尼亚大学

Berkeley 分校提供了资金，让他们在 UNIX 操作系统下实现 TCP/IP 协议。在这个项目中，研究人员为 TCP/IP 网络通信开发了一个 API（应用程序接口）。这个 API 称为 Socket 接口（套接字）。

20 世纪 90 年代初，由 Sun Microsystems、JSB Corporation、FTP software、Microdyne 和 Microsoft 等几家公司共同制定了一套 Windows 下的网络编程接口，即 Windows Sockets 规范。它是 Berkeley Sockets 的重要扩充，主要是增加了一些异步函数，并增加了符合 Windows 消息驱动特性的网络事件异步选择机制。

Windows Sockets 规范是一套开放的、支持多种协议的 Windows 下的网络编程接口。从 1991 年的 1.0 版到 1995 年的 2.0 版，经过不断完善并在 Intel、Microsoft、Sun、SGI、Informix、Novell 等公司的全力支持下，已成为 Windows 网络编程的事实标准。目前，在实际应用中的 Windows Sockets 规范主要有 1.1 版和 2.0.8 版。两者的最重要区别是 1.1 版只支持 TCP/IP 协议，而 2.0 版可以支持多协议。2.0 版有良好的向后兼容性，任何使用 1.1 版的源代码、二进制文件、应用程序都可以不加修改地在 2.0 规范下使用<sup>[25]</sup>。

## 2) Windows Sockets 通信机制

在计算机中，套接字（Socket）实际上提供了一个通信接口，应用程序在网络上传输和接收信息都通过这个 Socket 接口来实现，通过这个端口可以与任何一个具有 Socket 接口的计算机通信。在应用开发中，可以像使用文件句柄一样对 Socket 句柄进行读写操作。

Windows Sockets 支持两种类型的套接字，即流式套接字（SOCK\_STREAM）和数据报套接（SOCK\_DGRAM）。流式套接字提供了一个面向连接的、可靠的、数据无错的、无重复发送的及按发送顺序接收数据的服务。对于要求精确传输数据的 Windows Sockets 通信程序，一般采用流式套接字。数据报套接字提供供尽量传递的无连接数据报服务。对于每次传输少量数据或有实时需要的程序而言，适合采用数据报套接字。SOCK\_STREAM 和 SOCK\_DGRAM 传递数据的差异类似于电话和明信片之间的差异。SOCK\_STREAM 就像电话，必须先验证目标是否可以访问后才准备通信。SOCK\_DGRAM 就像明信片，信息量很小而且每次传递成功的可能性很高，但是

不能完全保证传递成功<sup>[28]</sup>。本网络通信程序的开发就非常适合采用数据报套接字来实现。因为下、下位机之间的数据传输有着数据量小，实时性高的特点。

## 3) MFC 对 Socket 的封装

MFC 中提供了 CSocket 类，其继承关系如下图 4-2 所示：

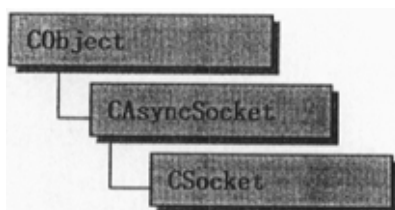


图 4-2 CSocket 类继承关系

MFC 对 Socket 的封装采用了如下实现步骤：

服务器	客户
//构造一个 socket CSocket sockSrvr;	//构造一个 socket CSocket sockClient;
//生成 SOCKET sockSrvr.Create (nPort);	//生成 SOCKET sockClient.Create ();
//进行侦听 sockSrvr.Listen ();	
	//实现链接 sockClient.Connect (strAddr,nPort);
//构造新的 socket CSocket socketRecv; //接受链接 sockSrvr.Accept (socketRecv);	
//构造文件对象 CSocketFile file (&socketRecv);	//构造文件对象 CSocketFile file (&sockClient);
//构造一个 archive 对象 CArchive arIn (&file,CArchive::load); 或 CArchive arOut (&file,CArchive::store);	//构造一个 archive 对象 CArchive arIn (&file,CArchive::load); 或 CArchive arOut (&file,CArchive::store);
//使用 archive 传送数据 arIn>>dwValue; 或 arOut<<dwValue;	//使用 archive 传送数据 arIn>>dwValue; 或 arOut<<dwValue;

## 4.2.3 Winsocket 的编程应用

本系统中，下位机每 20 毫秒调用一次定时中断将采集到的数据传送过来，其与上位机之间的数据传输量小，实时性强，并且每次传输不需要保证完全成功，所以在数据传送过程中不需要提供可靠的面向连接的传输服务。因此本设计采用数据报方式，并且是非阻塞方式的套接字。网间应用程序采用流行的 Client/Sever 模式。Winsocket 的编程通过 MFC 提供的 WinSock 类来完成。

MFC 对 Windows Socket API 进行了封装，提供了 WinSock 类 CSocket 和 CAsyncSocket。由于该类充分利用了 Windows 消息驱动机制，从而避免了直接采用 API 编程时必需另开一条线程进行侦听带来的多线程中的同步问题<sup>[29][30]</sup>。本程序通过从 CAsyncSocket 类派生出自己的 WinSock 类 CSocketSer，然后重载 CAsyncSocket 的虚函数 OnReceive (int nErrorCode)<sup>[31]</sup>。具体代码如下：

//套接字的构造与生成

```
m_pSocketSer = new CSocketSer (this); // 构造异步（非阻塞）套接字
```

```
int port = 5000; //初始化套接字端口号为 5000
```

```
if (m_pSocketSer->Create (port, SOCK_DGRAM, FD_READ | FD_WRITE, NULL)  
    == SOCKET_ERROR) //生成套接字为数据报方式
```

//重载 CSocketSer 的虚函数 OnReceive ()

```
void CSocketSer::OnReceive (int nErrorCode)
```

```
{
```

```
    m_pDoc->OnUdpReceive (); //文档类中的数据处理函数
```

```
    CAsyncSocket::OnReceive (nErrorCode);
```

```
}
```

//使用套接字接受数据

```
void CMonitorDoc::OnUdpReceive ()
```

```
{
```

```
    ...
```

```
int nRes;

nRes=m_pSocketSer->ReceiveFrom ((char*) &strRecv,sizeof (strRecv) ,
    strRecvIP, nRecvPort);    //m_pSocketSer 是在文档类中打开的套接字指针
if ((nRes != SOCKET_ERROR)  &&  (nRecvPort = 5000)) {...}

...

}
```

## 4.3 本章小结

本章详细讨论了串行通信接口程序的开发及网络通信程序的开发的原理及实现，介绍了在 Win32 环境下实现串行通信的方法以及需要注意问题，分析了 TCP/IP 协议组成以及通过 Socket 接口实现非阻塞方式通信的基本原理及步骤。



## 5 虚拟仪器软件的设计

快锻液压机组状态监测与故障诊断系统运用了虚拟仪器技术,采用面向对象技术开发的虚拟仪器是一个基于网络的虚拟仪表系统,主要用于对采集到的数据和状态信号进行实时监测和事后处理。能够用于开发虚拟仪器的软件开发平台很多,相对某些专用的虚拟仪器编程工具如 HP 的 VEE、NI 的 LabView 或是 LabWindows/CVI 而言,Microsoft 的 Visual C++ 6.0 无论在开放性和功能上都要优于后者。尤为突出的是,VC 使用户图形界面(GUI)的开发变得简单而迅速,在资金方面也比前者要经济。

Visual C++是一种可视化的面向对象的程序设计语言,具有很好用户界面。功能强大的 VC 提供的图形化控件拖拉式的设计,使界面设计更直观、方便。VC 还提供了简单高效的 ActiveX 控件开发工具,便于实现虚拟仪器软面板的组件化。Microsoft 提供给 Windows C++程序设计人员的一个面向对象的编程接口,其中 MFC 提供了 100 多个基类以及派生类的应用,从而在更高层次上实现代码的重用<sup>[32]</sup>。

### 5.1 软件框架结构

VC++支持的应用程序框架主要有基于文档/视图结构的应用程序(包括 MDI 和 SDI)和基于对话框的应用程序。表单视图 CFormView 是一种易于使用的视图,既支持文档/视图结构,又具有对话框特点。用户可以向表单视图的对话框资源中加入要使用的控件,并可方便地在应用程序与控件之间进行通讯,而且表单视图具有访问所有文档/视图接口的权限,可以直接访问文档类。通过文档/视图结构将数据及显示分开管理有利于用户数据的封装。本系统采用单文档-多视图结构,每一块虚拟仪器面板就是一个从表单视图 CFormView 派生的类,而测量数据类则作为文档类的成员,通过文档/视图接口的权限与面板对象进行交互,完成测量数据的实时动态显示<sup>[10][11]</sup>。

### 5.2 虚拟仪器面板设计

虚拟仪器是指具有虚拟仪器面板的个人计算机仪器,它是计算机资源、模块化功

---

## 5.2 虚拟仪器面板设计

虚拟仪器是指具有虚拟仪器面板的个人计算机仪器,它是计算机资源、模块化功能硬件与用于数据分析、过程通信及图形用户界面的应用软件的有机结合。在虚拟仪器中,软件是整个仪器功能的关键。这是因为其主要功能是由软件来体现的,即“软件就是仪器”。尽管“模块化”设计是实现虚拟仪器的指导思想,但传统的结构化技术已无法适应虚拟仪器中大量结构部件的组装和复杂并行的交互控制。由此提出了面向对象的解决方案。虚拟仪器的面向对象开发技术主要是采用抽象、封装、继承等办法来实现各种虚拟仪器部件,并用消息机制来实现虚拟仪器部件间复杂的交互作用和通讯。

虚拟仪器软面板是用户用来操作仪器,与仪器进行通信,输入参数设置,输出结果显示的用户接口。按照面向对象的设计思想,一个虚拟仪器集成系统由多个虚拟仪器组成,每个虚拟仪器均由软面板控制,而组成软面板的是大量的虚拟仪表<sup>[33][34]</sup>。

### 5.2.1 虚拟仪表

在虚拟仪表的可视化设计过程中,采用了面向对象方法。一个虚拟仪表系统由许多仪表组成,每个虚拟仪表都有各自的独立属性和行为,因此按照面向对象的设计思想将面板分解为具有继承关系的虚拟仪表,每个虚拟仪表单独设计成一个控件,可以方便的加入到派生自表单视图的面板上。在 VC 中制作自定义控件的方法很多,最好的方法是从现有的控件类派生。根据所需仪表的特点,这里选择 CStatic 作为基类。在 MFC 中,Windows 的设备环境(DC)被封装在 CDC 类中,CDC 类中包含绝大多数 GDI 调用。通过在重载的 OnPaint() 函数中使用 CDC 成员函数可以按要求绘制出逼真的仪表界面。使用这种方法制作的控件不仅可以很容易地改变大小、颜色等外观属性,而且可以通过成员函数定义很多的对象行为来控制控件的工作方式<sup>[35][36]</sup>。本系统通过这种方法制作了如图 5-1 所示棒图控件 C3DBarCtrl,图 5-2 所示趋势图控件 C3DTrendCtrl,图 5-3 所示罗盘控件 C3DMeterCtrl,图 5-4 所示滑块控件 C3DsliderCtrl,图 5-5 所示时钟控件 CXJWDigitClock 等。

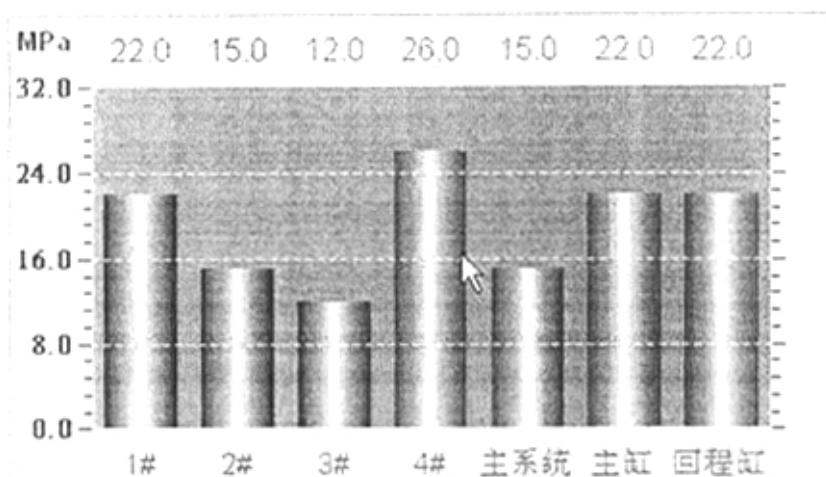


图 5-1 棒图控件

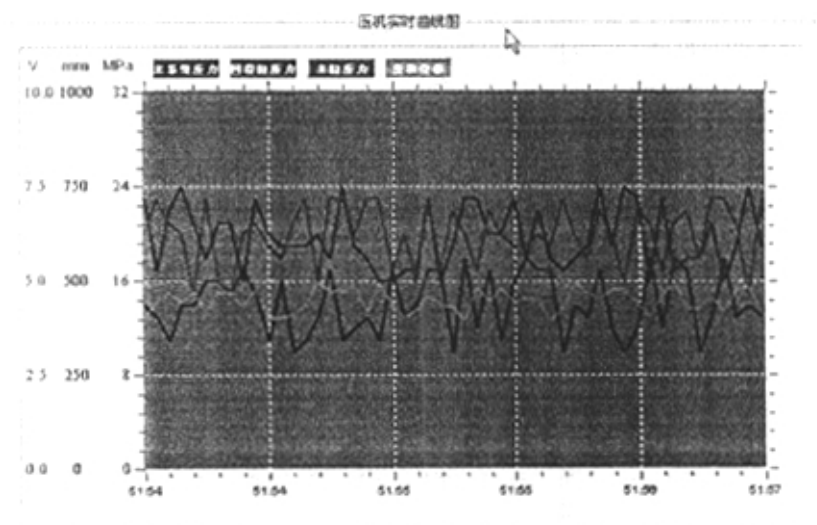


图 5-2 趋势图控件

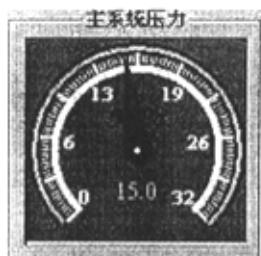


图 5-3 罗盘控件

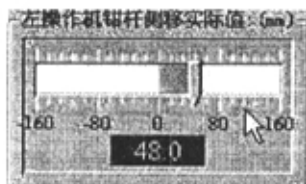


图 5-4 滑块控件



图 5-5 时钟控件

## 5.2.2 数据的动态图形显示

数据可视化 (Data Visualization) 就是将数据以特定的动态图形方式显示在计算机屏幕上, 供用户监控之用。在虚拟仪表中, 诸如罗盘的指针偏转和趋势图的波形变化等都反映了数据的动态变化。

在虚拟仪表的设计中, 当数据变化频繁时, 对应的仪表界面实时刷新将造成屏幕闪烁。一方面, 在MFC默认的屏幕绘图机制中, 总是先用背景色将显示区清除, 然后才调用OnPaint(), 而背景色往往与绘图内容反差很大, 这样在短时间内背景色与显示图形的交替出现, 使得显示窗口看起来在闪。要解决这个问题, 首先当然是去掉MFC提供的背景绘制过程了, 实现的方法很多: (1) 可以在窗口形成时给窗口的注册类的背景刷赋NULL。(2) 可以在形成以后修改背景 (3) 可以重载OnEraseBkgnd (CDC\* pDC) 直接返回TRUE。然后就要用到双缓存的方法了。双缓冲就是除了在屏幕上有图形进行显示以外, 在内存中也有图形在绘制。可以把要显示的图形先在内存中绘制好, 然后再一次性的将内存中的图形按照一个点一个点的覆盖到屏幕上去, 因为是非常规整的内存拷贝, 这个过程非常快。这样在内存中绘图时, 随使用什么反差大的背景色进行清除都不会闪。另一方面, 当数据变化时, 仪表界面中许多区域是固定不变的, 变化的只是与数据存在参数映射的部分区域。这里运用Windows动画编程的原理, 将固定不变的界面背景和要显示的图形分别在两块内存中绘制, 每次重绘仪表界面时, 如果背景没有变化, 则先将背景直接复制到将要显示的内存中, 再在这块已经有背景的内存中绘制动态变化的部分, 最后将其拷贝到屏幕上去<sup>[37][38]</sup>。下面列出了这个过程的主要实现代码, 其中用到了内存设备环境类CMemDC, 它派生自

CDC类，其析构函数完成从内存到屏幕的拷贝。

```
void C3DMeterCtrl::OnPaint ()
{
    CPaintDC dc (this) ;// 绘图设备环境
    GetClientRect (&m_rectCtrl) ;// 控件客户区大小
    CMemDC memDC (&dc, &m_rectCtrl) ;// 将要显示的内存设备环境

    // m_dcBackground 是绘制背景的内存设备环境
    // 如果背景还没有初始化或发生改变执行 if 语句
    if ((m_dcBackground.GetSafeHdc () == NULL) ||
        (m_bitmapBackground.m_hObject == NULL))
    {
        m_dcBackground.CreateCompatibleDC (&dc) ;
        m_bitmapBackground.CreateCompatibleBitmap (&dc,
            m_rectCtrl.Width (), m_rectCtrl.Height ()) ;
        m_pBitmapOldBackground=m_dcBackground.SelectObject (
            &m_bitmapBackground) ;

        // 以下语句绘制控件背景
        .....

    }
    // 将背景复制到将要显示的内存中
    memDC.BitBlt (0, 0, m_rectCtrl.Width (), m_rectCtrl.Height (),
        &m_dcBackground, 0, 0, SRCCOPY) ;

    // 以下语句绘制控件的动态部分
    .....
}
```

## 5.3 数据结构设计

快锻液压机组状态监测与故障诊断管理系统中涉及数据存储量大、数据类型多、数据处理次数多和对数据准确性要求高,其数据文件或数据结构的设计直接影响系统的处理效率和响应速度,并给用户使用和维护带来不便。

尽管随着计算机内外存储能力的增加,节省空间已不是重要问题。但数据文件或数据结构的合理设计,对提高系统的响应速度非常重要。为了合理设计数据结构,结合系统具体模块功能,分析它们的数据存储及处理的特点,为每个模块设计合理的数据数量和类型。MFC 的 Collections Class 中有许多适用于各种数据类型(如 Byte、Word、DWord、Prt)以及各种数据结构(如结构、数组、链表等)的现成类。同时,因为 MFC 提供的 CObject 类生来具有档案读写、动态生成等性质,从而方便数据的保存和处理。这里采用 CObject 类来派生自定义数据类,将采样数据作为自定义数据类的数据成员,将各种数据处理方法、数据表现方法都作为自定义数据类的成员函数。构造自定义数据类一般直接封装结构和数组,也有使用模板类的如 CArray 类、CList 等,它们支持复杂类的动态填加<sup>[1][39]</sup>。如果我们尽可能把这些现成的类应用在程序的数据结构设计上面,就可以节省许多开发时间。以下列出在系统中主要用到的几个数据结构,其它结构的设计与此类似,就不一一介绍了:

1) 实时数据类 CMonitorDataReal 派生自 CObject,用于记录每 64ms 从下位机传送到上位机中的所有参数,数据成员中包括一个实形数组 Data\_Real[],若干整形变量和 union(联合体),并提供函数 SetDataReal()用于各数据成员的设置。实时数据类用在 CMonitorDoc 文档类中用接受和暂存实时数据,并在各个视图类中起传递实时参数的作用。

2) 历史数据集类 CMonitorDataset 派生自 CObject,用于记录任意一个参数的最近 250 个值。为了保持这 250 值是最最近的按受到的,即能进行动态更新。该类采用两个 CArray 类,一个记录参数值,另一个记录每个值的时间信息。同时提供多个成员函数用于数据的增加,更新,获取和设置,以及串行化存取。历史数据集类是系统进行记录存取的主要类。

3) 曲线数据类 CTrendCurve 和曲线点信息类 CDataInfo 是专门用于复杂的趋势图控件的数据显示及处理而设计的。每个 CTrendCurve 类的实例保存任意一个参数的一条曲线信息，而曲线上每个点的信息又是由 CDataInfo 类的实例来表示。本设计中曲线数据类应用 CArray 类的 template 性质，指定数组内容为 CDataInfo 的指针，因为数组中的元素不是变量本身，而是其指针，所以在该类中元素的增删涉及到堆空间的分配和释放，所以要用到 new 和 delete 操作符进行内存分配。在成员函数 AddData() 中把通过 new 建立的 CDataInfo 对象作为数组元素加入到数组中，元素的删除则放到析构函数 ~CTrendCurve() 中用 delete 操作符释放全部 CDataInfo 对象。

## 5.4 故障数据库设计

故障数据库作为故障诊断功能模块中的基础部分，不仅满足系统高效可靠要求，还要为后续的功能扩展提供便利，所以故障数据库的建立与访问既要力求减少了软件开发的工作量，缩短开发周期，提高了效率和软件的可靠性，也要保证方法的灵活性与扩充性。基于这些考虑，我们采用 Access 数据库（即\*.MDB 文件）作为故障数据库，原因是其在办公自动化领域的通用性，同时也方便了系统管理人员应用其它通用数据库管理软件（如 Access、Visual FoxPro、Excel 等）打开该数据库进行分析和研究。而数据库的访问上则采用 DAO(Database Access Object)，数据库访问对象(DAO)提供了一种通过程序代码创建和操纵数据库的机制。多个 DAO 对象构成一个体系结构，在这个结构里，各个 DAO 对象协同工作。由于 DAO 是基于 Microsoft Jet 引擎的，因而在访问 Access 数据库（即\*.MDB 文件）时具有很好的性能<sup>[40][41]</sup>。

与 ODBC 一样，DAO 提供了一组 API 供编程使用。MFC 也提供了一组 DAO 类，封装了底层的 API，从而大大简化了程序的开发。利用 MFC 的 DAO 类，用户可以编写独立于 DBMS 的应用程序。以下是应用 DAO 编程的主要操作步骤：

1) 打开数据库 CDaoWorkspace 对象代表一个 DAO Workspace 对象，在 MFC DAO 体系结构中处于最高处，定义了一个用户的同数据库的会话，并包含打开的数据库，负责完成数据库的事务处理。我们可以使用隐含的 workspace 对象。CDaoDatabase 对象代表了一个到数据库的连接，在 MFC 中，是通过 CDaoDatabase



封装的。创建一个 CDaoDatabase 对象，而不明确地指定使用的 workspace,此时，MFC 将创建一个新的临时的 CDaoWorkspace 对象。代码如下：

```
m_pDB = new CDaoDatabase;  
m_pDB->Open (lpszPathName) ;// lpszPathName 是数据库完整路径名
```

2) 查询记录      一个 DAO recordset 对象，代表一个数据记录的集合，该集合是一个库表或者是一个查询的运行结果中的全部记录。CDaoRecordset 对象有三种类型：表、动态集、快照。通常情况下，我们在应用程序中可以使用 CDaoRecordset 的派生类，这一般是通过 ClassWizard 或 AppWizard 来生成的。本程序中我们从 CDaoRecordset 派生了 CDaoRecordsetAccess 类。

```
m_pSet = new CDaoRecordsetAccess (m_pDB) ;  
//以动态集类型打开查询结果  
m_pSet->Open (dbOpenDynaset,m_pSet->GetDefaultSQL (),0) ;
```

3) 添加记录      添加记录用 AddNew 函数，把记录定位到添加记录的位置，直接或用 SetFieldValue 来进行赋值。用 Update () 进行数据库中对应记录的更新。代码如下：

```
CDaoRecordsetAccess *pRS ;  
pRS = new CDaoRecordsetAccess (m_pDB) ;  
pRS->Open (dbOpenDynaset,pRS->GetDefaultSQL (),0) ;  
……//定位记录操作略  
pRS->AddNew () ;//添加记录  
pRS->m_Name="1" ; //赋值操作  
pRS->m_Notes="压机急停" ;  
m_strDataNote[0] = pRS->m_Notes;  
time = time.GetCurrentTime () ;  
pRS->m_Time=time;
```

pRS->Update ();//更新记录

.....

4) 修改记录      修改记录用 Edit () 函数, 把记录定位到要修改的位置, 调用 Edit 函数, 修改完成后, 调用 Update 函数。代码略。

5) 删除记录      删除记录用 Delete () 函数, 使用后不需调用 Update () 函数<sup>[42]</sup>。

## 5.5 本章小结

本章对虚拟仪器技术在计算机监测控制系统中的应用作了探讨, 介绍了如何在 Visual C++ 中制作虚拟仪表控件及虚拟仪器软面板的方法, 以及实现数据动态图形显示的技术关键, 最后还对软件中数据结构的设计及数据库的建立与管理作了简要介绍。

## 6 全文总结与展望

### 6.1 全文总结

本学位论文介绍了用 Visual C++ 开发基于虚拟仪表的状态监测与故障诊断系统的方法和技术,详细讨论了系统总体框架,数据通讯接口,虚拟仪表的软件设计思想和实现方法。开发出来的系统结构简单,界面美观,代码运行快,性能价格比高。现将本文所作的具体工作总结如下:

1) 在查阅大量国内外文献的基础上,探讨了计算机监测控制系统的软硬件组成及特点,完成了本系统的大部分设计工作。

2) 通过系统需求的认真分析,成功开发出了用监测计算机的快锻液压机组状态监测与故障诊断管理系统软件,对于采用面向对象技术开发监控软件作了有益的探索。

3) 通信是维持系统工作的基础,加上由于本系统特殊的性能要求,如跨平台,实时性等。使得设计高效可靠的通信接口程序成为必然。

4) 虚拟仪表技术是系统开发过程中的关键技术之一。为了能达到高速刷新无闪烁的目的,查阅了大量相关资料。终于完成了实用的虚拟仪表控件的开发。

实践证明该软件系统运行稳定,性能可靠,取得较好的社会和经济效益。

### 6.2 研究展望

快锻液压机组状态监测与故障诊断管理软件系统开发成功后,虽然已经初步实用化,在生产实际中发挥了其应有的作用,但是也暴露出很多局限性。比如缺乏故障诊断分析能力,系统的实时响应速度还有待加强,虚拟仪表的功能需要完善等。同时,由于系统处于初期开发阶段,部分功能需要进一步完善,另外还将增加许多更新更强的功能。针对以上问题,下面我就提出具体阐述一下后续研究的几个重点方向:

1) 运用面向对象技术开发基于知识的故障诊断专家系统,作为整个软件系统的

分析诊断功能模块。在故障数据库的支持下完成故障的归类整理及分析预测，进一步提高设备管理维修的效率。

2) 运用 ActiveX 技术制作全新的虚拟仪表控件，这些 ActiveX 控件将取代原有的自定义型普通控件，进一步提高控件的模块化水平和可移植性。使得软件代码更易维护和修改，功能则更易扩充<sup>[25][43][44]</sup>。

3) 提高软件在数据处理及存取上的智能化程度，尽量减少人工干预。

4) 进一步开发友好、直观、生动的人—机界面，达到快速及时，信息清楚、明确、充分的目的，确保工作人员的工作能力得到充分可靠地发挥。

# 华中科技大学硕士学位论文

---

## 致 谢

本论文是在导师陈柏金副教授、陈幼平教授的亲切关怀和悉心指导下完成的。导师学识广博、学术思想敏捷、治学态度严谨，不但具有总揽全局、胸怀开阔的领导风度及正直无私的人格魅力，而且具有严谨求实、开拓进取的治学风范，这一切无不是作者现在和将来学习的榜样。特别是导师对学生在获取知识、运用知识和创新三种能力方面的培养，使我受益非浅，并且将伴随着我人生旅程的每一个进步。在此文完成之际，谨向尊敬的导师致以诚挚的谢意，感谢导师为作者的成长所付出的心血，并送去学生最美好的祝愿！

在课题的研究过程中，得到了莫健华老师、叶春生老师、艾武老师、袁楚明老师、史玉升老师、钟绍辉老师、周钢老师等的大力协助，这几位老师不仅在工作上大力支持作者的研究，多次与作者进行有益的探讨，使作者少走了许多弯路，而且在生活上也给予了作者很多关怀。在此，作者向他们表示由衷的感谢！

作者在攻读硕士学位期间，得到了郑江硕士、刘骥刚硕士、蔡少虬硕士、夏强志硕士、丁静硕士、彭程硕士、李建华硕士、龚兴武硕士的合作和帮助，并一起度过了愉快的求学生活。在此，一并致以真诚的谢意！

作者要衷心感谢父亲、母亲和妹妹的鼓励和帮助，感谢他们给予作者的悉心关怀和支持！

感谢卢荣硕士在论文写作期间给予的理解和鼓励，她的支持也是激励作者前进的动力。

衷心感谢所有关心和帮助过我的朋友们！

作者：秦勇

2003年4月12日于沁园

## 参考文献

- [1] 赵林度. 大型机电系统故障诊断技术. 北京: 中国石化出版社, 2001
- [2] 刘计辉. 设备状态监测和诊断技术的现状与发展动向. 青海电力, 2000, (2): 50~53
- [3] 倪世青. 智能诊断系统的研制: [硕士学位论文], 华北电力大学, 2000
- [4] 逢振旭, 倪其民, 李从心等. 快速锻造操作机的计算机控制. 锻压机械, 2000, (1): 53~54
- [5] 逢振旭, 李从心, 倪其民等. 快速锻造液压机的结构与控制. 上海交通大学学报, 2000, 34 (10): 39~40
- [6] Arif M, Ishmara T, Inooka H. Iterative learning control using information database [J]. Journal of Intelligent and Robotic System, 1999, 25 (2): 27~41
- [7] 张健成, 周恩涛, 曹走铭等. 2000t 快锻机的在线监测与维护. 冶金自动化, 1997, (4): 17~19
- [8] 胡友民, 杜润主, 杨叔子. 液压系统运行状态监测. 液压与气动, 2002, (8): 35~37
- [9] 黄世琦, 丁勇, 谢志江. 一种基于 DOS/Windows 的在线监测与故障诊断系统. 重庆大学学报 (自然科学版), 1997, 20 (5): 95~98
- [10] David J. Kruglinski, Scot Wingo, George Shepherd. Programming Microsoft Visual C++6.0 (5<sup>th</sup> ed.). Microsoft Press, 1999
- [11] 侯俊杰. 深入浅出 MFC (第二版). 武汉: 华中科技大学出版社, 2001
- [12] John E. Swanke. VC++ MFC Extension by Example. Mill Freeman Inc, 1999
- [13] 钱能. C++程序设计教程. 北京: 清华大学出版社, 1999
- [14] Bjarne Stroustrup. The C++ Programming Language(3<sup>rd</sup> ed.). Addison Wesley Longman, 1997
- [15] 杨厚宗. 信息化时代设备状态监测与故障诊断技术的新特点和发展趋势. 中国

# 华中科技大学硕士学位论文

---

设备工程, 2002, (11): 37~39

- [16] 周德泽, 袁南儿, 应英. 计算机智能监测控制系统的设计及应用. 北京: 清华大学出版社, 2001
- [17] Harmon J M, Napoli J. Realtime diagnostics improve power plant operations. *Power Engineering*, 1992, 96 (11): 45~46
- [18] Smith D J. Monitoring/diagnostic systems enhance plant asset management. *Power Engineering*, 1992, 96 (6): 23~29
- [19] 张安华. 机电设备状态监测与故障诊断. 西北工业大学出版社, 1995
- [20] 屈梁生, 陈岳东. 计算机辅助监测与诊断技术. 西安: 西安交通大学出版社, 1989
- [21] 李现勇. Visual C++ 串口通信技术与工程实践. 北京: 人民邮电出版社, 2002
- [22] Jan Axelson. *Universal Serial Bus Complete*. Lakeview Research, 1999
- [23] [美]Mark Nelson. 串行通信开发指南 (第二版). 潇湘工作室译. 北京: 中国水利水电出版社, 2000
- [24] 张力. Visual C++高级编程. 北京: 人民邮电出版社, 2002
- [25] 胡晓军, 邓波, 高宏伟等. Visual C++高级开发范例解析. 北京: 电子工业出版社, 2002
- [26] Douglas E. Comer. *Internetworking with TCP/IP Vol. I: Principles, Protocols, and Architecture (3<sup>rd</sup> ed.)*. Prentice Hall, Inc, a Simon & Schuster company, 1995
- [27] Anthony Jones, Jim Ohlund. *Network Programming for Microsoft Windows*. Microsoft Press, 1999
- [28] M. F. Arnett, M. Coulombe. *Inside TCP/IP (2<sup>nd</sup> ed.)*. New Riders, a Simon & Schuster company, 1996
- [29] 贾青, 谢剑英. 采用 WINSOCK 设计 Windows 下的网络异步通讯. *自动化与仪表*, 1998, 13 (3): 53~55
- [30] 陈雷. 用 MFC 类库实现 Winsock 编程. *计算机应用研究*, 1997, (4).
- [31] 刘颀, 陈仕伟, 胡亚非. 用 MFC 实现煤矿主通风机监测数据的网络通讯. 风



# 华中科技大学硕士学位论文

---

机技术, 2001, (3): 41~43

- [32] 韩慧莲, 徐晓东, 张伦. 虚拟仪器软件平台设计. 华北工学院测试技术学报, 2002, 16 (1): 1~3
- [33] 杨亚东, 赵世廉. 图形化虚拟仪器软面板的设计. 电子技术, 2000, (11): 49~51
- [34] 王晓霞. 虚拟仪器, 仪表的一种新的发展模式. 计算机自动测量与控制, 1997, (4)
- [35] 汪红. 虚拟示波器的软件实现. 中南民族学院学报(自然科学版), 2000, 19 (1): 35~38
- [36] 吉国力, 杨炬会, 陈伟. 虚拟仪器的面向对象开发技术. 厦门大学学报(自然科学版), 2001, 40 (6): 1202~1206
- [37] 顾郁莲, 蔡宣平, 颜飞翔. 虚拟仪表的可视化技术. 电子技术应用, 2000, (6): 29~31
- [38] C. H. Small. Virtual Instruments, EDN, 1998
- [39] John E. Swanke. VC++ MFC Extension by Example. Miller Freeman, Inc, 1999
- [40] Jeffrey D. Ullman, Jennifer Widom. A First Course in Database Systems. Prentice Hall, Inc, 1999
- [41] Chuck Wood. OLE DB and ODBC Developer's Guide. IDG Books Worldwide, Inc, 1999
- [42] 郑章. Visual C++ 6.0 数据库开发技术. 北京: 机械工业出版社, 1999
- [43] 彭莉辉, 吴鸿修. 基于 ActiveX 的多通道数据曲线编辑控件的实现. 计算机应用研究, 2000, (1): 102~103
- [44] Warren Ernst. PRESENTING ACTIVEX. Sams. net Publishing, 1996
- [45] 郭颖, 王兆玲. 大机组状态监测故障智能诊断系统的设计及应用. 炼油与化工, 2002, 13: 46
- [46] 黄志坚, 戴稳兴, 蔡明清. 珠钢轧机液压在线监测系统. 南方钢铁, 2000, 114 (6): 13~17

## 附录 1 攻读硕士期间发表论文目录

- [1] 秦勇, 陈柏金. 基于虚拟仪表的状态监测与故障诊断系统. 仪表技术, 2002, 187  
(6): 30~32 (署名单位: 华中科技大学)
- [2] Chen Baijin, Qin Yong, Zheng Jiang, Huang Shuhuai. SIMULATION AND  
EXPRIMENT OF HYDRAULIC SERVO SYSTEM OF LARGE FLUX  
ELECTRO-HYDRAULIC PROPORTIONAL VALVES: Li Zhuangyun, Zhu Yuquan,  
Li Baoren, eds . PROCEEDINGS OF THE FOURTH INTERNATIONAL  
SYMPOSIUM ON FLUID POWER TRANSMISSION AND CONTROL  
(ISFP'2003). WUHAN, CHINA. APRIL 8-10, 2003. Beijing: Beijing World  
Publishing Corporation, 2003: 483~488