

基于 Web 的网络实验远程监控系统的设计

马 林, 霍大勇, 薛丁箫

(河南工业职业技术学院电气工程系, 南阳 473009)

摘 要: 从系统集成的角度研究了基于 Web 的远程监控系统设计中所涉及的关键技术, 设计了基于 Web 的远程实验系统, 实现了本地监控站的设计和远程客户端与 Web 服务器的设计。

关键词: 网络结构; 设备服务器; 控制算法; 程序设计; 客户端

Design of Network Experiment Long-distance Supervisory System Based on Web

MA Lin, HUO Dayong, XUE Dingxiao

(Dept. of Electrical Engineering, Henan Polytechnic Institute, Nanyang 473009)

【Abstract】 This paper thoroughly studies the essential technology from the system integration angle which involves Web based on long-distance supervisory system design, designs a long-distance experimental system based on Web, and implements the local monitoring station design and the long-distance client side and Web server design.

【Key words】 Network structure; Device server; Check algorithm; Programming; Client side

1 网络实验室系统结构

随着自动控制技术、计算技术、网络技术的迅猛发展, 实现生产过程控制的高效率、高精度、高柔性、高集成化已成可能并成为重要的发展方向。为此进行了大胆的尝试并达到了预期的满意的目标。即开发了基于 Internet 远程监控实验系统, 它突破了时间和空间的限制, 通过 Web 界面可以远程进行真实监制。整个实验室的网络结构如图 1 所示。

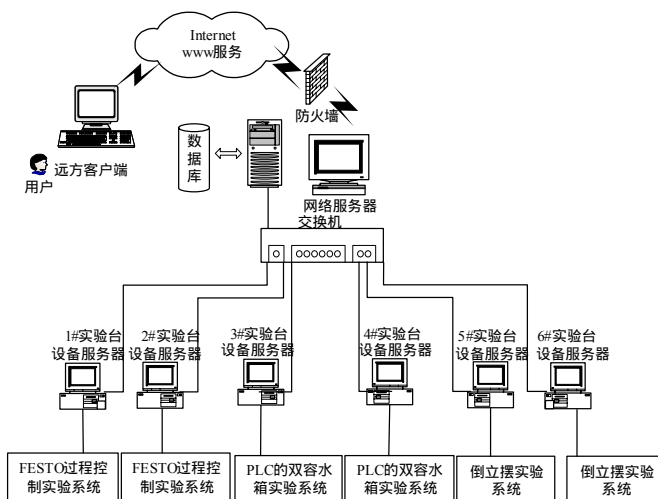


图 1 实验室网络结构

系统设计采用了多层结构, 各部分的功能如下:

(1) 设备服务器模块

设备服务器即现场监控站, 接受来自网络服务器验证的远方客户端请求, 负责执行远方用户的操作代理请求, 同时在本地实现对实验系统的控制算法, 并且把远程客户实验参数和实验历史数据存储到数据库中, 允许远方用户通过动态网页查询历史数据。

(2) 网络服务器模块

管理登录用户, 协调各个实验台的分配, 负责对实验的管理、实验信息的发布更新、某些插件(如 Java Plug-in)和参考源程序(控制程序)的下载。

(3) 远程用户模块

远程用户模块采用 ActiveX 或 Java Applet 嵌入到网页中的形式, 实现远程用户同本地实验台信息的交互, 所有的用户操作都将在这个模块中完成, 通过与设备服务器的通信实现实验过程, 同时与网络服务器通过动态网页交互实现用户注册、登录。

2 设备服务器模块的设计与实现

设备服务器模块的软件设计主要包括 3 部分内容: 设备服务器与 PLC 站的通信, 控制算法, 远程监控代理设计。

2.1 设备服务器与 PLC 站的串口通信

本地水位监控系统采用 Omron 公司的 C200HG 系列的 PLC 为下位机, 用 RS-232 型电缆将 Host Link 模块直接连到作为设备服务器的计算机串口上。本文采用 C++Builder6.0 作为开发设备服务器与 PLC 的串口通信程序。

(1) PLC 与上位机连接系统的通信协议

上位机与 PLC 之间的通信实际上是计算机与 PLC 通信模块 Host Link 之间交换命令和响应的过程。上位机具有初始传送优先权, 所有通信均由上位机启动, 不需要 PLC 编写通信程序, Host Link 能够对上位机发送来的字符串进行分析, 检查数据格式, 分析指令代码, 然后根据指令代码进行相应的操作, 并向上位机发出响应信号, 通知上位机已完成或反映通信的错误, 如奇偶校验错误、FCS 错误、代码错误等。

作者简介: 马 林(1963 -), 女, 实验师, 主研方向: 电气技术实验; 霍大勇, 高工; 薛丁箫, 副教授

收稿日期: 2006-05-21

E-mail: malin20055@yahoo.com.cn

在一次交换中传输的命令格式和应答数据称为一帧。命令帧要通过用户编写的上位机通信程序实现，PLC 的上位链(HostLink)单元会根据上位机发来的命令帧自动生成响应帧返回给上位机。

(2)用 C++Builder 开发 PLC 与设备服务器串口通信程序

上位机串口编程采用 Visual Basic 提供的 MSCComm32 控件来实现。在进行开发时，安装 Visual Basic 后，则在系统中自动注册一个 ActiveX 串口控件 MSCComm32，在 C++Builder 环境中将其导入，之后就可以在 C++Builder 中进行串口程序开发时使用该控件。

1)PLC 上位机链接设置

采用 RS-232C 端口时，需要置 DM6645 的 12-15 位为 0，PLC 的节点号设置在 DM6648 的 00-07 位。本文采用 OMRON C200 PLC 的默认设置，即在 CPU 的 DIP 开关 J 脚置 OFF 的情况下，在 PLC 与设备服务器之间采用如下参数进行通信：启动位：1 位；数据长度：7 位；停止位：2 位；奇偶校验：偶校验；波特率：9 600bps。

2)设备服务器中串口通信程序的设计

通过串口通信程序实现 PLC 与设备服务器之间的数据交换，其步骤如下：初始化串口，设置双方通讯的端口号、波特率、数据位、停止位、奇偶校验方式、打开串口、输入/输出缓冲区清零等工作。

整个串口的通信程序流程如图 2 所示。

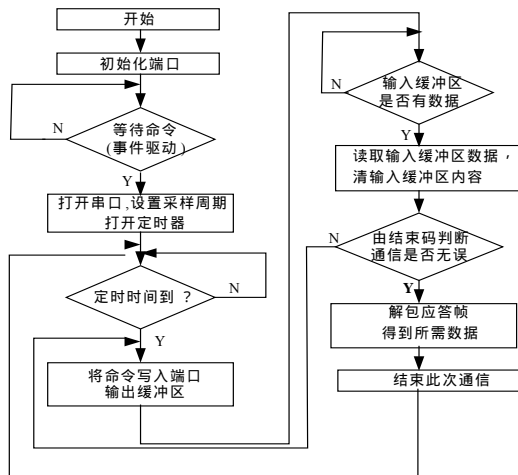


图 2 设备服务器与 PLC 串口通信流程

2.2 实验控制算法的实现

双容水箱实验平台在设备服务器上完成对现场水位的控制，远方客户端可以进行控制参数的设定。整个控制算法的设计可以在远方客户端实现，实现真正的远程控制。为了实现远程用户也能参与控制算法的设计，并且能通过现场装置验证用户控制算法运行的效果，在进行控制算法设计时，我们提供了二次开发接口，主要是 DLL 技术来实现控制算法的动态加载。用户可以使用 Visual Basic、Visual C++、Delphi 或 C++Builder 编程工具按照指定的函数名称和参数类型编写对应的接口函数，编译成 DLL 即可。然后通过 FTP 上传到设备服务器指定的路径中，并且在进行参数传送时告诉设备服务器采用默认控制算法还是用户自定义控制算法。用户只需着重于控制程序的编写，因此对于用户来说，实验的复杂性没有太大的提升，而系统的灵活性却大大地提高了。

(1)利用 DLL 技术编写控制算法

所谓动态链接是把一些经常会共享的代码制成 DLL 文

件，当运行文件调用到 DLL 文件内包含的函数时，Windows 系统才会把 DLL 文件加载到内存中，如果在 Windows 系统上同时运行好几个应用程序，而这些应用程序又都使用到共同链接库，那么 Windows 系统将只保留一份代码在内存中。当应用程序调用到某个函数时，DLL 的运行机制是：Windows 系统先检查内存中是否存在该函数代码，如果有，就直接调用它，如果没有，Windows 系统会把此函数对应的 DLL 文件加载到内存中，再进行链接。DLL 机制的一个最大特点就是语言兼容性，无论是用 Visual Basic、Visual C++、Delphi 还是 C++Builder 编写的 DLL 文件，上述任何语言都可以调用。

下面讨论采用 C++Builder 开发 DLL 文件实现串级 PID 算法的过程。

首先是建立 DLL 模板程序，在 C++Builder 开发环境中选择 [File]→[New]→[Other] 功能，打开对象库，然后点击 [New] 选项卡中的 DLL Wizard 图标，再点击 [OK]，在 DLL Wizard 对话框中选择 C++，C++Builder 将自动建立模板程序。

采用串级 PID 算法的系统结构如图 3 所示。

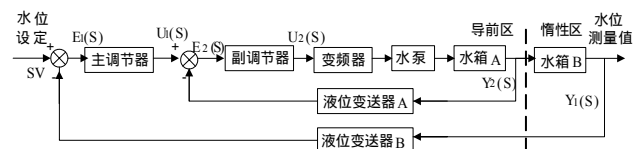


图 3 串级控制系统结构

其控制算法公式如下：

$$e_1(k) = SV - y_1(k)$$

$$\Delta u_1(k) = K_{c1}[e_1(k) - e_1(k-1)] + K_{i1}e_1(k) + K_{d1}[e_1(k) - 2e_1(k-1) + e_1(k-2)]$$

式中， $K_{i1} = \frac{K_{c1}T}{T_i}$ 称为积分系数；

$$K_{d1} = \frac{K_{c1}T_{d1}}{T} \text{ 称为微分系数。}$$

$$u_1(k) = u_1(k-1) + \Delta u_1(k)$$

$$e_2(k) = u_1(k) - y_2(k)$$

$$u_2(k) = K_{c2} \left\{ e_2(k) + \frac{T_{d2}}{T} [e_2(k) - e_2(k-1)] \right\}$$

(2)调用 DLL 形式存在的控制算法文件

在程序中调用 DLL 文件有两种方式：静态调用和动态调用。静态调用时，DLL 在程序开始执行的时候就自动载入，其占用系统资源较多；而动态调用方式仅仅在需要的时候才将 DLL 调入内存，这样能够减少系统资源的占用，而付出的代价是调用过程有些复杂。使用静态调用除了需要 DLL 文件外，还需要导入对应的库文件 (*.lib)，而动态调用时只需要 DLL 文件，考虑到系统要实现远方客户自定义控制算法，在设计中采用动态调用方式。

2.3 远程监控代理程序的设计

远程监控代理程序是实现远程客户和现场设备进行信息交换的“桥梁”，远程监控代理程序跟远方客户的通信采用 Socket 技术来实现，监控代理程序作为 Socket 的服务器端，通过与远程 Socket 客户端的通信，实现在 Internet 上传输数据。同时把用户设定的操作参数和现场实时数据存入数据库中，供用户进行本地和远方查询。

在远程网络监控软件系统设计中，应用层数据传输协议的设计是一项很重要的工作。只有采用统一的数据传输协议，远程客户端和本地监控站才能“理解”对方发送数据的含义，

(下转第 274 页)