

# 基于 Internet 的闸门远程监控技术及实现

束长宝, 陈兴, 史旺旺, 蒋步军

(扬州大学信息工程学院, 扬州 225009)

**摘要:** 根据水利工程中地理位置分散的闸门监控特点和功能, 设计和开发了基于 Internet 的远程监控系统, 该系统集现场监控、远程监控、视频、管理于一体。介绍了实现闸门远程监控的关键技术及其核心程序: 如基于 Win32 API 函数的串行通信、采用 CAsyncSocket 类的 socket 网络通信和视频信号处理等。

**关键词:** 闸门; Internet; 远程监控

## Technology of Remote Monitor for Sluices Based on Internet and Its Realization

SHU Changbao, CHEN Xing, SHI Wangwang, JIANG Bujun

(Institute of Information Technology, Yangzhou University, Yangzhou 225009)

**【Abstract】** Remote supervisory control system for sluices based on Internet is designed and implemented, according to the distributing feature and function of sluices in hydraulic engineering. The system is integrated with field control, remote supervisory control, video and management. The key technology and essential programs are introduced such as serial communication based on API function, socket communication by CAsyncSocket class and process of video signal.

**【Key words】** Sluices; Internet; Remote monitor

闸门控制的自动化是合理调配水资源、提高利用率的有效手段。水利系统中的闸门地理位置往往比较分散、环境恶劣, 要求无人值守, 用传统的控制方法不便管理。基于 Internet 的闸门远程监控系统, 采用计算机技术、通信技术、测控技术、视频技术、自动化技术等, 是实现闸门远程监控系统测控管一体化的一种有效方法。

江苏省某市水利局的闸门远程监控系统基本情况如下: 黄村闸有 3 孔平板闸门, 3 水闸为 1 孔平板闸门, 两地分别距离水利局约 1km 和 3km, 承担着该市内河水位控制及引水冲污控制任务。系统实现闸门开度、上下游水位检测, 并根据闸门流量模型计算过闸流量; 实现闸门开度和流量的自动和手动控制; 实现闸门运行情况远程实时图像监控; 两闸运行数据与上级部门网络互连, 以便及时了解闸门运行情况。

### 1 系统网络结构

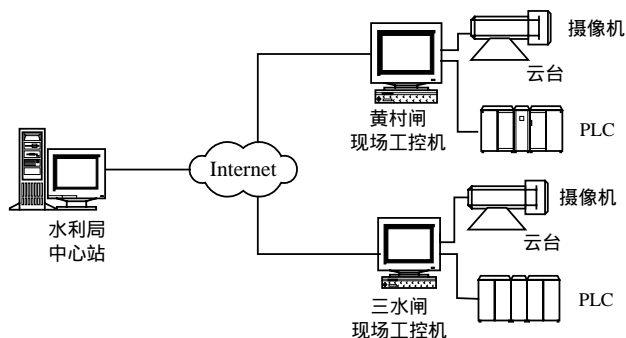


图 1 系统网络结构

根据水利局中心站、黄村闸及 3 水闸测控站地理位置分布特点, 单独架设光缆成本高, 系统采用租用电信局光缆、

并分配固定 IP 地址的方法联结 3 地的网络通信。将闸门自动控制系统和视频监控统系统统一设计, 减少重复投资。采用的系统网络结构如图 1 所示。

### 1.1 主要硬件及功能

根据现场情况及控制要求, 硬件配置如下:

(1) 现场监控单元: 选用性价比较高的 OMRON CPM1A-40CDR 可编程控制器(PLC)及 MAD01 模拟量扩展单元, PLC 通过 CIF01 通信适配器与上位工控机 RS-232C 口连接。

(2) 现场工控机: 采用研华 IPC-610 工控机, 内装视频采集卡和以太网卡, 兼做服务器。通过 PLC 对闸门高度、上下游水位数据进行实时监测, 通过视频采集卡和相应算法实现对视频信息的采集, 处理后通过宽带网将数据传输到水利局中心机房的监控主机; 同时接收来自后者的控制指令, 通过 PLC 实现闸门现场监控和摄像机、云台控制等。

(3) 局中心站监控主机: 作为客户机, 是远程监测中心, 也是测控管一体化网络系统的核心, 由连在水利局局域网上的工控机实现。系统通过 ODBC 技术实现与 SQL Server 数据库连接, 以供远程用户访问。

### 1.2 开发软件

本系统以控制为主, 集信息管理与视频监控为一体。考虑到实时性要求, 选择速度较快的开发平台。系统开发软件:

- (1) 系统平台: Windows 2000 profession, IIS 5.0;
- (2) 数据库: SQL Server 2000;

**基金项目:** 江苏省教育厅自然科学研究指导性计划基金资助项目 [KK0410182]

**作者简介:** 束长宝(1969 - ), 男, 硕士、讲师, 主研方向: 电气自动化, 计算机控制等; 陈兴, 硕士、讲师; 史旺旺, 硕士、副教授; 蒋步军, 学士、实验师

**收稿日期:** 2006-06-06 **E-mail:** scbyz@tom.com

- (3)开发平台：VC++ 6.0；
- (4)PLC 编程及调试软件：CX-Programmer 2.0。

## 2 具体实现

### 2.1 基于 Win32 API 函数的串行通信

OMRON PLC串行通信采用HOST Link协议,其本质为串行通信,上位机与PLC通信过程中数据以“帧”为单位传送,上位机一般具有优先权并启动通信,发命令帧给PLC,PLC收到后自动发响应帧。上位机可以通过读写PLC的DM区或IR区的信息等实现数据采集和输出控制功能,数据采用FCS帧校验方式<sup>[1]</sup>。

本文采用使用Win32 API函数实现对RS232 串行通信口的访问,虽程序设计较复杂,但控制灵活、响应速度快<sup>[2~4]</sup>。利用微机RS-232 口进行串行通信的一般步骤为串口初始化、启动线程、进行超时设置和数据读写、关闭串口<sup>[5]</sup>。串行通信流程见图 2。

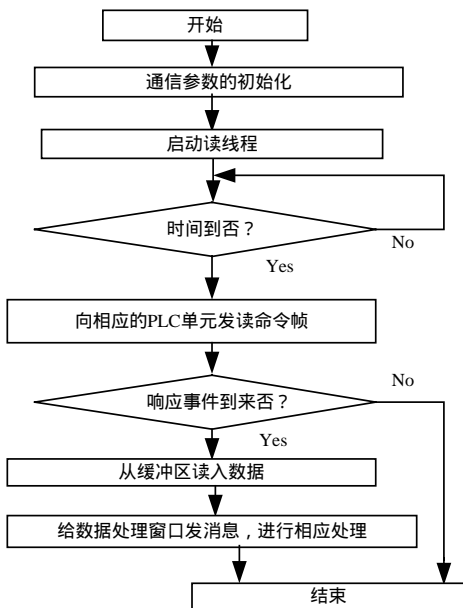


图 2 串行通信流程

#### (1)串行口的初始化

通信时先通过函数 CreateFile()以异步读写方式打开串口。串口打开后通过调用 GetCommState(m\_hCom,&dcB)读取当前串口设备控制块 DCB 的设置,设置内容主要有串口号、波特率、数据位、有无奇偶校验、停止位和事件字符,这些参数应与 PLC 中设置相同。通过 SetCommState(m\_hCom,&dcB)将修改后的串口设置写入,调用 SetupComm()设置通信缓冲区大小,从监控系统可靠性角度考虑,再通过相关函数设定串口通信超时参数、清除接收和发送缓冲区等。

#### (2)启动接收和发送线程

本系统使用 Win32 应用程序中提供的工作线程,用于监视串口事件,为后台采集数据。串口打开后,通过函数 AfxBeginThread()启动读线程。

#### (3)读写数据

读线程内的读数据过程在自定义的函数 ReadComm()中实现,在调用 ReadFile()读串口前,先调用 ClearCommError()确定接收缓冲区内的数据量,并在此基础上加 1,并延时 1ms,等待帧结束符\*后的回车符的到来,这样可读取完整的一帧。具体如下:

```
length=min(dwLength,ComStat.cbInQue+1); //确定读缓冲区字节
```

```
//数,加 1 读回车键
if(length>1)
{
    Sleep(1);
    fReadStat=ReadFile(m_hCom,buf,length,&length,&m_osRead);
    if(!fReadStat)//读出数据量小于希望的数据量,返回 0
    {if(GetLastError()==ERROR_IO_PENDING)//等待结果中
        {GetOverlappedResult(m_hCom,&m_osRead,&length,TRUE);
        //确定实际读取数量
        }
    }
}
```

在接收线程中,用 WaitCommEvent()函数判断事件的发生:

```
if(!WaitCommEvent(pRecv->m_hCom,&dwEvtMask,&os))
{
    if(GetLastError()==ERROR_IO_PENDING)
    {
        GetOverlappedResult(pRecv->m_hCom,&os,&dwTrans,TRUE);
        if((dwEvtMask&EV_RXFLAG)==EV_RXFLAG)
        //事件发生
        {
            length=pRecv->ReadComm(pRecv->pNextRecvChar,MAXBLOC
            K); //读缓冲区
            PostMessage(pRecv->m_hTermWnd,pRecv->m_Msg,
            EV_RXFLAG, pRecv->m_dTotalRecvLength);
        }
    }
}
```

读完完整的一帧数据后,用 PostMessage()向指定窗口的消息队列发送通知消息,相应地在该窗口建立消息与成员函数间的映射,用 ON\_MESSAGE 将消息与成员函数名关联。

写数据利用 VC++中的定时控制,定时向 PLC 发送读数据命令,可直接调用 WriteFile()函数。

(4)数据读完之后关闭串口,释放串口资源。

### 2.2 WinSock 网络通信

Windows Sockets API是Microsoft Windows的网络程序设计接口,它使在Windows下开发高性能的网络通信程序成为可能<sup>[6,7]</sup>。Windows Sockets API被封装在控件WinSock中。用WinSock进行网络通信,有两种主要方式:流方式和数据报方式。流方式采用TCP协议,通信可靠,本系统中闸门监控信息交换采用该方式。数据报方式采用UDP协议,无重发校验机制,通信速率较高,视频信息的传输采用此方式实现。

socket通信采用MFC WinSock的类进行网络通信。Visual C++的MFC类库提供了 2 个Socket类:一个是CAsyncSocket基本类,它提供全面的由事件驱动的Socket通信能力,在较低的级别上封装了Windows Sockets的API,灵活性好;二是Csocket类,它是CAsyncSocket派生的子类,在较高层次上提供了一种编程手段<sup>[6,7]</sup>。考虑到实时性和灵活性要求,本文采用CAsyncSocket类通信方式。

系统程序包括服务器端程序、客户端程序两部分:

#### (1)服务器端程序设计

对服务器端,首先从 CAsyncSocket 派生一个新类 CServerSocket,然后用这个新类定义 CAsyncSocket 对象,即

