

一个电源监控网络系统的分析与设计

许金水 陈世平 (上海理工大学网络管理中心 200093)

摘要	本文分析了一个电源监控网络系统的总体结构,介绍了EMAC通信协议、Socket通信、Windows下对串口的操作,以及该系统支局监控软件、总局监控软件和基于Web界面的查询模块的实现。
关键词	电源监控 EMAC Socket通信 串口通信

1 前言

上海金山电信局有三十几个支局端,电源设备分布于各支局端,而支局端与总局之间的距离少则五六公里,多则三十几公里。以前,每个支局端必须要有各自的电源监控设备及人员来处理,造成了人力、财力的浪费,而且不利于集中监控、集中管理、集中查询等。针对这一情况,该电信局决定重新布置网络,用光纤将各支局与总局联网。本文讨论了金山电信局电源监控系统的研制与开发。

2 系统总体结构

本系统采用网络技术,将各个支局的信息汇报到总局(控制中心),并可将控制中心的控制命令下发到各支局。

控制中心:对各个支局的信息进行存储、查询和打印;对网络进行维护;对用户进行管理。

支局:接收电源设备的信息,并向控制中心发送其状态信息。支局具有信息的临时存储功能,在网络出现故障恢复正常后,原本正常的支局可自动恢复连接。

系统总体结构如图1所示。

其中,EMAC(Equipment Monitor And Control)为电力设备监视控制装置的简称。EMAC用来监视在各支局的电力设备(例如电源),将电力设备的警报信号、计测信号等模拟信号转换成数字信号并通过RS232传送到上位通信集约装置(这里是PC)。EMAC与PC间的通信有专门的通信协议——EMAC通信协议。支局端PC与总局端PC采用TCP/IP协议进行通信。总局端安装有Web Server及DataBase Server,客户端采用WEB浏览器界面。总局端与客户端采用Browser/Web Server/DataBase Server模式,

系统管理员不用再为客户操心,无需在客户端安装查询软件,只要其有浏览器,且是处于联网状态,无论何时何地都可进行查询。

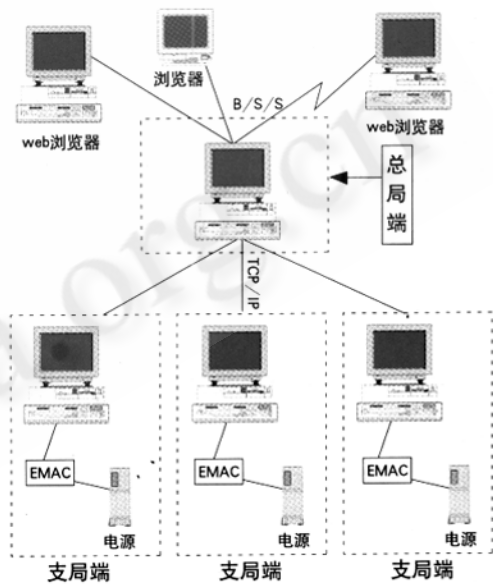


图1 系统拓扑结构

本系统管理软件包括支局监控软件、总局监控软件、基于Web界面的查询模块三部分。

2.1 支局监控软件

支局监控软件负责接收来自EMAC的信号,进行判断是否是正常数据,如果是正常数据,则向总局端发送正常数据,同时将正常数据显示在本地机的屏幕上;如果不是正常数据,则产生告警数据,并向总局端发送,同时将告警数据显示在本地机的屏幕上。

2.2 总局监控软件

总局监控软件负责接收各支局的数据,并提供管理、

查询等功能。总局按要求显示支局的正常数据及报警数据，并将正常数据及报警数据存入数据库以便查询。每个支局用一个线程，支局监控软件与总局监控软件之间通过 Socket 通信。

电源监控系统分为三大模块：系统管理模块、查询模块、帮助模块。

系统管理模块：打开监控、关闭监控、交接班、告警声音屏蔽、告警确认、用户管理等。

查询模块包括：查询历史常规数据、查询历史告警数据、查询日志。查询的结果提供打印功能。

帮助模块：提供整个系统的使用说明。

2.3 基于 Web 界面的查询模块

本模拟使得管理人员可以在任何地方，任何时间使用浏览器进行历史常规数据、历史告警数据及日志的查询，大大方便了管理。

3 通信协议

3.1 EMAC 通信协议

主EMAC提供RS232接口与上位通信集约装置相联，主EMAC与副EMAC之间采用RS485标准。

信息传送方式为异步方式，起始位1位，数据位8位，停止位1位，无校验。

(1)上位通信集约装置与主EMAC之间的特性代码

① 控制代码：使用JISX0201规定的传送特征代码以及以下特征代码：

特征代码	代码	特征代码	代码
STX	0X02	ETX	0X03
EOT	0X04	ENQ	0X05
ACK	0X06	选择时序	0X1B
ETB	0X17		0X03

② 信息代码（包括BCC）：对于控制代码以外的信息代码，将8位数据的最高位置为1，以示区别。

③ 地址代码（0X80~0X8F，可选择16台）：从上位来的轮询(ENQ)及选择(1B-ENQ)后附加有地址。

(2)上位通信集约装置与主EMAC之间的通信协议

上位通信集约装置与主EMAC之间的通信遵守“三次握手”协议，图2表示的是上位通信装置在向上EMAC传送信息的情况下，完成初始化启动过程后，对主EMAC进行选择时序通信。

① 上位通信集约装置方

B：“ENQ”接收不到“ACK”时，在将“1B-ENQ”送出后2秒内，接收不到“ACK”时，将“1B+ENQ”再送出。如果连续4次送出后也接收不到“ACK”时，将回路切断。

E：“EOT”接收不到“ACK”时，在将“要求电文”送出后2秒内，接收不到“ACK”时，将“1B+ENQ”再送出。如果连续次送出后也接收不到“ACK”时，将回路切断。

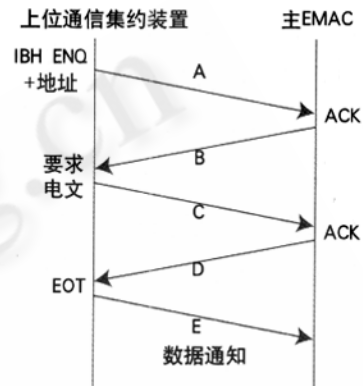


图2 选择时序通信协议

② 主EMAC方

B：不可接受ACK时，在“ACK”送出后，2秒以内从上位通信集约装置侧接收不到“STX”时，回到初始状态。

另外，“要求电文”出现BCC错误时，回到初始状态。再者，当来自上位的轮询地址与EMAC本身的地址不同时，不进行响应。

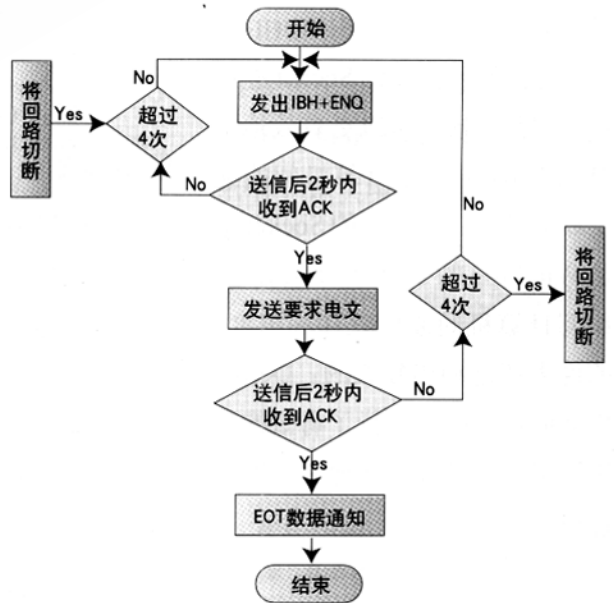


图3 上位通信集约装置通信图

3.2 支局端与总局端之间的通信

支局端与总局端之间的通信采用基于 TCP/IP 协议 Socket。

计算机网络是由一系列网络通信组成的，其中的核心协议是传输层的 TCP/IP 协议。TCP 是面向连接的，通信双方保持一条通路，好比目前的电话线，使用 telnet 登陆 BBS，用的就是 TCP 协议；UDP 是无连接的，通信双方都不保持对方的状态，浏览器访问 Internet 时使用的 HTTP 协议就是基于 UDP 协议。TCP 和 UDP 协议都非常复杂，尤其是 TCP 协议，为了保证网络传输的正确性和有效性，必须进行一系列复杂的纠错和排序等处理。

Socket 是建立在传输层协议(主要是 TCP 和 UDP)上的一种套接字规范，最初是由美国加州 Berkley 大学提出，它定义两台计算机间进行通信的规范(也是一种编程规范)，如果说两台计算机是利用一个“通道”进行通信，那么这个“通道”的两端就是两个套接字。套接字屏蔽了底层通信软件和具体操作系统的差异，使得任何两台安装了 TCP 协议软件和实现了套接字规范的计算机之间的通信成为可能。

微软的 Windows Socket 规范(简称 WinSock)对 Berkley 的套接字规范进行了扩展，利用标准的 Socket 的方法，可以同任何平台上的 Socket 进行通信；利用其扩展，可以更有效地实现在 Windows 平台上计算机间的通信。在 Delphi 中，其底层的 Socket 也应该是 Windows 的 Socket。Socket 减轻了编写计算机间通信软件的难度，但总的说来还是相当复杂的；Delphi 中对 Windows Socket 进行了有效的封装，使得用户可以很方便地编写网络通信程序。

4 电源监控系统的实现

4.1 支局监控软件

支局监控软件涉及的技术主要

有：对串口的操作、多线程编程。

(1) 对串口的操作。Windows 操作系统的机制禁止应用程序直接访问计算机硬件，但它为程序员提供了一系列的标准 API 函数，使得应用程序的编制更加方便并且免除了对有关硬件的调试。在 Windows95/NT 中，原来 Windows3.X 的 WM_COMMNOTIFY 消息已被取消，操作系统为每个通信设备开辟了用户可定义大小的读/写缓冲区，数据进出通信口增多由操作系统后台完成，应用程序只需对读/写缓冲区操作即可。WIN95/NT 中几个常用的串行通信操作函数如下：

CreatFile:打开串行口
CloseHandle:关闭串行口
SetupComm:设置通信缓冲区的大小
ReadFile:读串口操作
WriteFile:写串口操作
SetCommState:设置通信参数
GetCommState:获取默认通信参数
ClearCommError:清除串口错误并获取当前状态

① 利用 CreateFile 函数打开串行口，以确定本应用程序对此串口的占有权，并封锁其他应用程序对此串口的操作；

② 通过 GetCommState 函数填充设备控制块 DCB，再通过调用 SetCommState 函数配置串口的波特率、数据位、校验位和停止位；

③ 创建串行口监视线程监视串行口事件。在此基础上就可以在相应的串口上操作数据的传输；

④ 用 CloseHandle 函数关闭串行口。

(2) 多线程编程。在 WIN95/NT 中进行串行通信除了解基本的通信操作函数外，还要掌握多线程。线程是进程内部执行的路径，是操作系统分配

CPU 时间的基本实体。每个进程都由单线程开始完成应用程序的执行。串行通信需要利用多线程技术实现，其主要的处理逻辑可以表述如下：进程一开始先由主线程做一些必要的初始线程发送 WM_COMMMESSAGE 消息(自己创建)，主线程对其进行处理。若不需要 WM_COMMMESSAGE 消息，则主线程终止通信监视线程。具体实现如下：

① 自定义消息： WM_COMMMESSAGE=WM_USER+100；

② 在适当位置对自定义消息处理过程(MsgCommProcess)进行声明；

```
Procedure MsgcommProcess(Var Message:TMessage); Message WM_COMMMESSAGE;
```

③ 当指定的串行口事件发生时，向主线程发送 WM_COMMMESSAGE 消息；

④ 实现自定义消息处理过程。

4.2 总局监控软件

总局端需安装 Web Server(IIS) 及 Data Base Server (SQL Server)，以实现 Web 发布、执行 ASP 脚本及数据库服务。总局监控软件通过 Socket 与支局监控软件通信，接收来自各支局的正常数据和告警数据，把它们显示在屏幕上，产生相应的告警，并把正常数据和告警数据存入数据库以备查询和打印。总局监控软件还具备用户的管理(增加、删除、修改等)，登入该系统需密码验证，并将上班、下班、告警确认等情况存入数据库以备查询和打印。

4.3 总局与支局之间通信的实现

Delphi5.0 下，网络通信组件一般是工作在 TCP/IP 协议基础上，可以利用这些组件设计基于 Client/Server 模式的应用程序，通常，用于通信的组件总是成对的，如 ClientSocket 和

ServerSocket, NMMsg 和 NMMSGServer 等。本系统采用 NMMsg 和 NMMSGServ 组件, 主要目的是通过 Internet 或 Intranet 发送简单的 ASCII 码文本信息。无论是客户端程序还是服务器程序, 都要使用 NMMsg 和 NMMSGServ 两个组件, 任一方要发消息, 则调用该方的 NMMsg 组件的有关方法和事件, 发送完毕即断开连接; 接收消息则调用该方的 NMMSGServ 组件的有关方法和事件, 它监听 TCP/IP 网络上其他计算机的连接请求, 收到请求时就建立连接, 连接成功即接收消息, 接着又继续监听其他客户端的连接请求, 因此, 可以同时建立多个连接, 实现与多客户的通信。

当支局端接收来自 EMAC 的数据, 进行处理以后, 调用 NMMsg 组件的 PostIt 方法将数据发送到总局端, 部分程序如下:

```
iniConfig:=TiniFile.Create  
(GetiniFileName);
```

```
//从ini文件中读取总局端的IP地  
址
```

```
NMMsg1.Hos:=iniConfig.  
ReadString('设备情况','主机地  
址,");
```

```
//从ini文件中读取支局名,随数  
据发送,让总局端知道数据由哪个支  
局发来的
```

```
NMMsg1.FromName:=iniConfig.  
ReadString('设备情况','支局  
名,");
```

```
//将数据信息 DataInformation 发  
送到总局端
```

```
NMMsg1.PostIt(DataInfor-  
mation);
```

当总局端接收到来自支局端的数据时, 将触发 NMMSGServ 组件的 OnMsg 事件, 在 OnMsg 事件中, 对数据进行相应的处理, 例如, 显示

在屏幕上, 存入相应的数据库等。

4.4 基于 Web 界面的查询模块的实现

本系统采用了 Browser/Web Server/DataBase Server 模式, 采用了基于 WINNT+IIS+ASP 的技术解决方案, 系统使用 SQL SERVER 数据库。ASP 是基于 WinNT 和 IIS 的开放式脚本开发环境, 它把 HTML 和脚本语言很好地融合在一起, 它与数据库的交互是可以用 ADO 或 ODBC 等。利用 ASP 可以很容易地实现动态、交互式、高效地 Web 应用程序。■

参考文献

- 1 吴宛萍、黄求根, *Delphi4.0 中的网络消息传递机制及应用*, <http://www.chinainfo.gov.cn/>。
- 2 蒋东兴等, *Windows Socket 网络程序设计大全*, 北京清华大学出版社, 1999。