

文章编号 :1671-7848(2006)03-0268-04

# 工业以太网和 CAN 总线系统的通信软件设计

姚晓峰, 陈晓侠, 张春光

(大连交通大学 电气信息学院, 辽宁 大连 116028)



**摘要:** 为了实现企业设备控制层到数据管理层的信息集成, 采用工业以太网和现场总线混合控制网络, 可以实现控制和管理的有机统一。采用 Maxim 公司推出的 8051 家族中的微控制器 DS80C400 芯片, 利用微型因特网接口(Tiny Internet Interfaces - TINI)平台, 采用 JAVA 语言, 应用套接字技术、OPC 技术和 JAVA 多线程技术, 完成了工业以太网和 CAN 总线网络间嵌入式网关的设计, 给出了两层网络间通信软件设计的方法。方案以 CAN 总线为底层控制网络, 以太网作为上层管理网络, 中间为嵌入式网关, 网关实现了异质网络之间的无缝连接。

**关键词:** 工业以太网; CAN 总线; 嵌入式网关; 套接字

中图分类号: TP 311

文献标识码: A

## Design of Communication Software for Ethernet and CAN Bus System

YAO Xiao-feng, CHEN Xiao-xia, ZHANG Chun-guang

(College of Electrical and Information, Dalian Jiaotong University, Dalian 116028, China)

**Abstract:** To realize the information integral of the device control layer and the data management layer, mixed network of Ethernet and Field-bus is introduced to realize the unification of control and management. The chip of Maxim DS80C400(8051 family) is adopted. By using Tiny Internet Interfaces(TINI)Platform and, JAVA Language with the technique of socket, OPC and multithreading, the design of embed gateway is given. The method uses CAN network as the bottom control net, and Ethernet as the manage net. The embed gateway realizes the perfect conjunction of different network.

**Key words:** ethernet; CAN bus; embed gateway; socket;

### 1 引言

现场总线有着分散控制更彻底、开放性好、可靠性高、总体价格比较经济等优点, 但由于现场总线在短期内难以形成一个单一的国际标准, 使其开放性受到限制。与此同时, 以太网由于其开放性好、应用广泛以及价格低廉备受关注。采用以太网和现场总线混合控制网络, 使工厂的管理可以深入到控制现场, 即通过 Intranet 远程监控工业生产过程和进行远程系统调试、设备故障诊断等<sup>[1]</sup>。针对这种情况, 本文设计一种单独的 CAN-以太网互连系统, 在大连某小区锅炉控制系统中得到成功应用, 实现以太网与 CAN 总线网络的直接数据互联, 实现异质网络之间的无缝连接。

### 2 基于工业以太网的现场总线控制系统的总体结构<sup>[2]</sup>

锅炉控制系统总体结构, 如图 1 所示。

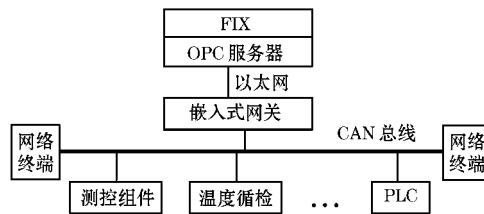


图 1 系统总体框图

在本系统中, 以测控组件、温度循检和 PLC 为智能节点, 其外接模拟量、数字量输入输出, 主要用于进行锅炉生产数据(如温度、压力、流量等)的检测, 上位机采用 FIX 监控软件, 上、下位机通过 OPC 服务器、嵌入式网关、CAN 总线进行通信, 完成数据采集与监控。

测控组件、温度循检和 PLC 功能由下载到智能节点上的组态序列决定, 具体阐述其软件体系由组件模块库、实时数据库和组态序列组成。

当某一组件模块响应系统呼叫为系统服务时, 首先从数据库中的指定单元输入数据, 进行相应的

操作处理后送回到数据库中。因此，由大量组件模块形成的组件模块库可完成各种各样的特定功能和数据操作。组态序列成为组件模块之间的消息序列。是基于“组件模块加组态序列构成软件解空间”的思想，并利用组态软件进行复杂的组态工作，例如对于测控组件其中与通信有关的两行行组态程序为

```
50, 1, 13, 9, 0, 0, 100;
51, 2, 20, 14, 10;
```

对于第一行，50 代表功能号，表示下位机向网桥传送数据，1 表示工程量编号(传送数据时以 0-1 小数传送,工程量决定数据的实际值)，13, 9 表示要传送的数据下位机的哪个数据单元，两个 0 与组态语法有关，在此不做赘述，100 代表要传送到网桥缓冲区的地址，此行组态程序完成的功能即是下位机数据库 13, 9 里的数据传送到网桥地址为 100 的缓冲区地址中。对于第二行，51 代表功能号，表示下位机接收上位机上传的数据，2 表示工程量编号，20 代表要传送到下位机的缓冲区地址，14, 10 表示要把数据存到下位机的哪个数据单元。由此看以，无论上传或下传数据，都有相应的缓冲区地址作为标识，而此地址与 OPC 服务器中配置的 OPC Item 的 PATH 相对应。

系统模拟量采集点 110 个，数字量采集点 20 个，模拟量输出 12 个，数字量输出 18 个。

综上分析可见系统通信任务的实现关键是实现 CAN 协议的数据与以太网数据的相互转换，即嵌入式网关软件部分的设计。

### 3 嵌入式网关软件的设计

嵌入式网关软件主要完成 OPC 服务器和智能测控组件的通信，本文中现场总线选择 CAN2.0B 协议的 CAN 总线，以太网使用 TCP/IP 协议，应用层基于 Modbus/TCP 协议，下面详细介绍协议转换的设计思路与实现方式。

1) CAN2.0B 通信协议 CAN 总线是一种串行数据通信协议。CAN2.0B 使用 29 位的扩展报文标识。一个数据帧由 7 个不同区域组成，如图 2 所示。

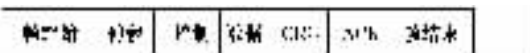


图 2 CAN 数据帧结构

执行 CAN 协议的 CAN 芯片可由多家制造商提供，本文使用的是 Philips 半导体公司生产的 XJA1000(支持 CAN2.0B)CAN 芯片。

2) TCP/IP 协议和 Modbus/TCP 通信协议 CAN2.0B 与 TCP/IP 通信协议都没有定义应用层标

准，而应用层协议的选择直接关系到系统的开放性与互操作性。由于本文是面向工业控制领域，所以应用层协议的选择还应充分考虑其可靠性与实时性，当然，价格与可操作问题也是非常重要的。综合上面的因素，本文选择 Modbus/TCP 作为应用层协议。其数据帧格式如图 3 所示。

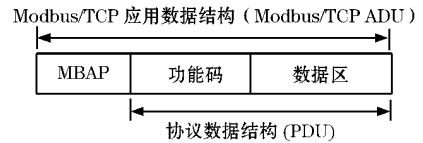


图 3 Modbus/TCP 数据帧格式

3) 硬件支撑平台 本文采用 DS80C400 芯片来完成网关的开发，DS80C400 芯片内集成了 CAN2.0B 现场总线控制器、1-Wire net 总线控制器和遵从 IEEE 802.3 协议的以太网媒体接入控制器。并具有完整的 TCP/IP 协议栈，DS80C400 芯片提供最大的系统时钟频率为 75 MHz，片内 ROM 固化了微型因特网接口(Tiny Internet Interfaces - TINI)平台，该平台是由一个基于微控制器的芯片组和支撑固件组成，功能强大、使用方便。网关的硬件部分在 TINI 平台上，配合相应的硬件及软件开发工具，可以开发并调试具备 CAN 总线底层控制网络和基于 Ethernet 顶层管理网络的总线控制系统。

4) 软件结构及设计思想 通信软件结构如图 4 所示。

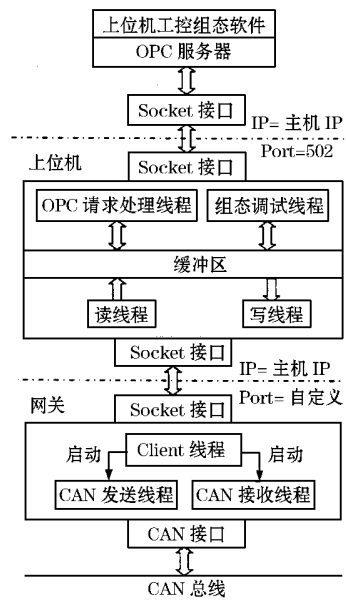


图 4 嵌入式网关软件结构

整个通信软件可分为两大部分，一部分为网关部分程序，一部分为上位机部分。在网关部分，CAN 发送线程负责将上位机的组态数据和控制数据通过 CAN 总线下载到挂接在 CAN 总线上的智能节

点, CAN 接收线程主要功能是操作 CAN 控制器, 从底层现场总线网络中接收数据(即底层控制网络的智能设备采集的实时数据或报警信息), 并将数据存储在缓冲区, 实时刷新缓冲区中的存储内容。在上位机部分, 其中读、写线程负责与网关部分进行数据通信, 而 OPC 请求处理线程负责通信软件对 OPC 服务器的请求处理及响应, 组态调试线程用来对下位机进行功能组态及参数调试等功能。

5) 通信软件方案的实施 TINI 开发平台支持 JAVA 语言、汇编语言、C 语言, 考虑到 JAVA 语言在网络通信方面具有其不可比拟的优越性, 本文采用 JAVA 语言开发网关。开发平台除了具有 JAVA 通用的工具包外, 还固化有专门操作 CAN 总线和以态网的工具包, 工具包中提供方法对 CAN 总线操作相当方便。下面以上传数据为例介绍通信方案的具体实施, 上传数据时协议转换的设计思路为:

a. 接收 CAN 数据帧, 拆包后提取数据区数据与上传的地址信息; b. 将数据和地址封装成 Modbus/TCP 数据帧的数据区内; c. 通过 TCP/IP 协议栈将 Modbus/TCP 数据帧封装在 TCP 数据包中; d. 通过以太网发送数据。

①收 CAN 数据帧 首先从 CAN 总线接收上传的数据帧, 其部分 JAVA 程序如下:

```
canbus.receive( frame );    接收 CAN 帧
data = frame.data;         提取数据
id = frame.ID;             提取 ID 号
framelength = frame.length; 提取帧长度
```

网关应用程序与上位机应用程序的通信

②两个程序之间进行双向数据传输 用 Socket 接口实现比较方便, 本应用系统中用到了两个 Socket 接口, 一是网关部分 CAN 发送线程和 CAN 接收线程与上位机部分的读写线程进行通信时所用到的, 另一个是上位机 OPC 服务器的请求处理线程、组态调试线程与 OPC 服务器通信时所用到的。以前一个 Socket 为例, 其建立过程如下。

在服务器端(上位机)部分代码如下:

```
ServerSocket socket = new ServerSocket ( 999 )
```

调用 ServerSocket 类以 999 端口号为参数, 创建一个 ServerSocket 对象, 即是服务器端的服务程序在该指定端口进行监听。

```
Socket client ; client = socket.accept ( );
```

服务器端程序使用 ServerSocket 对象的 accept() 方法, 接收来自客户机程序的连接请求, 此时服务器端将一直保持停滞状态, 直到收到客户端发来的连接请求。

使用新建的 Socket 对象创建输入、输出流对

象。使用流对象的方法完成和客户端的数据传输, 按约定协议识别并处理来自客户端的请求数据, 并把处理的结果返回给客户端。

在客户端(网关)部分代码如下:

```
socket = new Socket( ip , port );
input = socket.getInputStream( );
output = socket.getOutputStream( );
在服务器端部分代码如下:
input = client.getInputStream( );
output = client.getOutputStream( );
is = new DataInputStream( input );
os = new DataOutputStream( output )
```

在此处应注意的是上位机应用程序通过 Socket 接口从网关应用程序获取 CAN 协议帧, 提取数据、地址等信息后, 必须按 Modbus/TCP 数据帧格式重新组织数据, 发送给上位机 OPC 服务器, 以作为 OPC 服务器提出的数据请求。

③针对 OPC 服务器的快速应答 OPC 服务器处于网关与监控软件的接口, 对 OPC 服务器请求的应答速度对整个系统性能影响至关重要。下面是通过实验获得的 OPC Server( Modbus/TCP )读写整形量的请求与响应格式。

请求格式:

```
0 1 00 06 2 4 00 01
```

响应格式(十进制格式):

```
0 1 00 05 2 4 2 0 16
```

OPC Server 是遵循 Modbus/TCP 协议的, 其响应由若干字节组成, 其中第 2, 3, 4, 7 字节在 OPC 配置完成后每次响应都是保持不变的。因此, 响应数据包的组成过程中不必每次都对每一个字节进行操作的, 而是在定义组成响应数据包字节数组时对第 2, 3, 4, 7 字节初始化, 从而简化响应过程, 提高响应速度, 保证在正常数据量的情况下, OPC 服务器单个数据项的响应速度在 0.02 s 以下。

下载数据与上传数据方向相反, 原理相同。

④通信软件特点及优势 软件采用多线程技术保证了数据读、写及下载调试等多任务需求, 开辟缓冲区可以显著提高系统实时性能, 减少线程间的耦合, 增强程序的可扩展性及复用性。OPC 技术提高了控制系统的互操作性和适应性。

## 4 结 语

本文介绍了基于以太网和现场总线控制系统的通信软件设计方案。该互连方案保证了生产测控层与管理监控层之间的可靠连接。系统在大连某小区供热系统中得到了成功的应用。(下转第 273 页)