

CAN 总线与以太网互连系统设计

摘要: 介绍了一种基于单片机 S X 5 2 的 C A N 与以太网互连方案, 阐述了以太网和 C A N 总线网络协议转换的软硬件设计, 实现了以太网与现有 C A N 总线网的直接连接。保证管理监控层 (以太网) 与生产测控层 (C A N 总线网) 之间的连接, 使得上下层数据能方便地通信。

关键词: 现场总线 CAN 总线 以太网 隔离滤波器 YL18-1001D

在大型企业自动化系统中, 上层企业管理层和生产监控层一般都采用以太网和 P C 机, 而下层车间现场则采用现场总线和单片机测控设备。上下两层的沟通, 通常采用工业控制机加以以太网卡, 再加上 P C 机插槽上的接口卡或并行打印口的 E P P 接口卡实现。这种连接方式成本高, 开发周期长。针对这种情况, 笔者设计一种单独的 C A N 以太网网关互连系统, 成功地实现以太网与现有 C A N 总线网的直接数据互联。

1 系统结构

系统总体结构分为三部分: 现场测控网络 (C A N 网络)、嵌入式透明 S X 5 2 网关、以太网信息管理终端 (如监控平台和网络数据库等), 如图 1 所示。

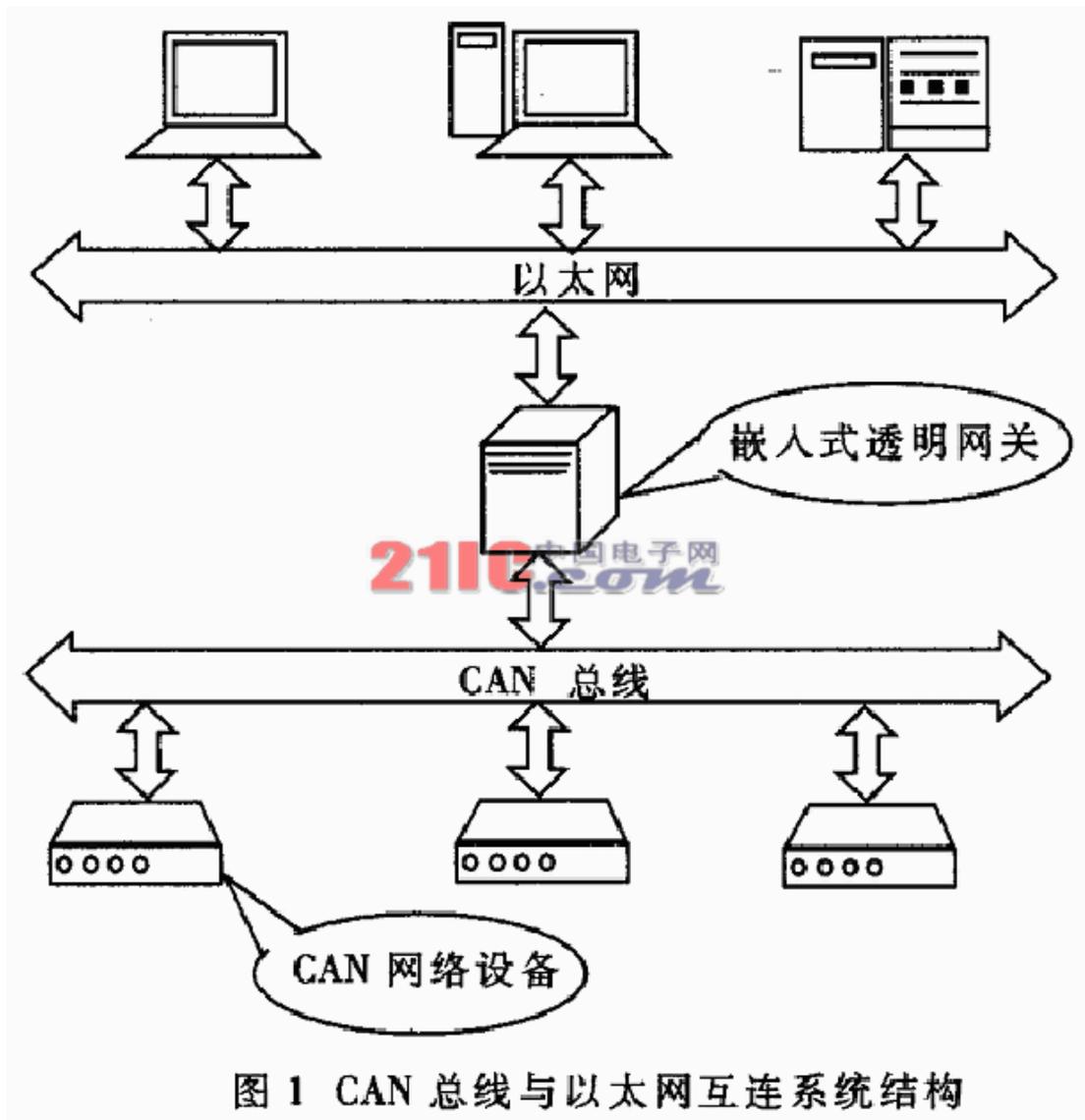


图 1 CAN 总线与以太网互连系统结构

C A N 总线是一个设备互连总线型控制网络。在 C A N 总线上可以挂接多达 1 1 0 个设备节点, 各设备间可以自主相互通信, 实现复杂网络控制系统。但设备信息层无法直接到达信息管理层, 要想设备信息进入信息管理层需通过数据网关。嵌入式透明 S X 5 2 网关就是为此而设计的。

透明式网关在以太网应用层构建和解析完整的CAN协议数据包。CAN协议数据包作为TCP/IP网络应用层的数据进行传输，它对通信数据的具体实际意义不做任何解释。透明式网关由通信处理器、CAN总线控制器和以太网控制器三部分组成。其中SX52单片机为核心处理器，它实现了CAN控制网络与以太网之间的协议转换。以太网信息管理层的控制指令发送到嵌入式透明SX52网关，将TCP/IP协议包数据转换为CAN协议形式发送至CAN控制网络中的指定设备节点，完成信息管理层对现场设备层的控制。同样地，当CAN网络上的设备数据（如定时采样数据或报警信息）要传输到信息管理层时，可将数据发送到嵌入式透明SX52网关，再通过网关协议转换程序将CAN协议数据封装成TCP/IP协议的以太网数据帧发送至以太网上的监控计算机。

以太网信息管理终端是一个根据用户的具体要求而设计的用户层应用软件。它可以是一个WIN32监控程序或网络数据库（记录CAN节点设备数据）软件等；甚至可能是CAN节点设备的服务器软件，为设备提供较复杂的数据处理工作。

2 硬件设计

系统硬件分为两大部分：CAN总线网络设备接口设计和嵌入式透明SX52网关设计。

2.1 CAN总线网络设备接口设计

CAN总线网络设备接口设计较网关设计简单。它是在完成设备功能的基础上加入一个CAN通信控制器接口芯片，实现与CAN总线网络的连接。考虑到开发成本和灵活性，笔者在设计中选用PHILIPS公司的独立CAN通信控制器SJA1000芯片和CAN总线收发器82C250芯片。其结构如图2所示。

2.2 嵌入式透明SX52网关设计

嵌入式透明网关设计是整个系统的核心。其结构如图3所示。它由CAN控制器协议转换模块和以太网控制器协议转换模块两部分组成。网关硬件中SX52微处理器起核心作用。它是由美国Uicom公司研制的高速可配置通信控制器，其处理速度相当高。在外接100MHz时钟时，指令执行速度可达100MIPS。它可实现TCP/IP协议栈中的ARP、IP、UDP、TCP、HTTP、SMTP、ICMP等网络协议。

CAN控制器协议转换模块硬件电路原理如图3左框图。它由三部分组成：微控制器SX52、独立CAN通信控制器SJA1000、CAN总线收发器82C250。其中SX52为唯一的CPU核心，负责SJA1000的初始化，通过读写SJA1000内部寄存器实现数据的接收、发送和错误处理等。

PCA82C250则提供对总线的差动发送能力和对CAN控制器的差动接收能力。

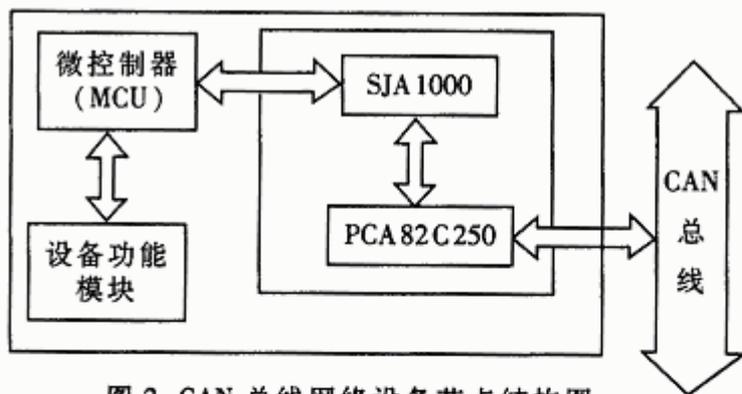


图2 CAN总线网络设备节点结构图

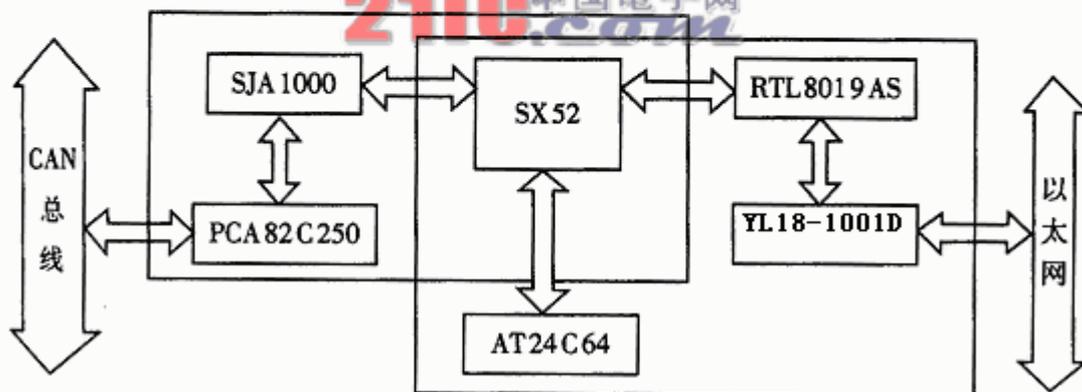


图3 嵌入式透明SX52网关结构图

以太网控制器协议转换模块主要由微控制器SX52、以太网通信控制器RTL8019AS和隔离滤波器YL18-1001D组成。RTL8019AS是台湾Realtek公司制造的一种高集成度的全双工10Mbps以太网控制芯片，实现了基于Ethernet协议的MAC层的全部功能，内置16KB的SRAM、双DMA通道和FIFO完成数据包的接收和发送功能。在网关设计中，使用跳线模式（JP置为高）硬配置RTL8019AS为8位模式。使用RTL8019的低5位地址线A0~A4以及低8位数据线D0~D7。SX52的B口的B0~B4脚作为地址线连接RTL8019AS的低5位地址线，B5~B7作为控制线分别连接读写时序控制脚IORB、IOWB、IOCHRDY；C口作为数据线连接RTL8019AS的低8位数据线；A口保留，用作日后扩展。图3中AT24C64为8KBEEPROM，主要用来保存嵌入式透明SX-52网关的配置信息，如网关IP地址、MAC地址和SJA1000的ID网络标示符、网络掩码AMR和总线定时（BTR0、BTR1）等。这样可以灵活方便地修改网关参数，适应不同环境，同时也考虑到以后的扩展。

RTL8019AS除与SX52连接外，还将其网络收发器的4根引脚TPOUT+、TPOUT-、TPIN+、TPIN-通过外接的隔离滤波器YL18-1001D与以太网相连。采用隔离滤波器YL18-1001D是为了提高网络通信的抗干扰能力。

3 软件设计

整个互联系统的软件设计可以分为三部分：CAN总线设备接口通信程序、透明网关协议转换程序和以太网层应用程序设计。其中，CAN总线设备接口通信程序和透明网关协议转换程序的CAN控制器协议模块在结构上有较大的相似性，但有可能因采用微控制器不同而导致实现的程序语言相异。因而，在此不作论述，而主要讨论后两个方面的程序设计。

3.1 透明网关协议转换程序

透明网关协议转换程序的整体设计思路为：当以太网应用层有数据要发送到CAN节点时，首先，数据发送到透明网关由以太网控制器协议转换模块从传输层数据报文中解析出完整的CAN协议数据包，存放在

数据缓冲区A 再通知总调度模块，由它调用CAN控制器协议模块将CAN协议数据包发送到CAN总线上。反过来，当CAN设备有数据要发送到用户层时，首先，数据发送到透明网关由CAN控制器协议模块将完整的CAN协议数据包存放在数据缓冲区B 再通知总调度模块，由它调用以太网控制器协议转换模块将完整的CAN协议数据包作为应用层数据封装起来，再发送到以太网的应用层。其程序结构如图4所示。

3.1.1 CAN控制器协议模块

CAN控制器协议转换模块程序主要由SJA1000的寄存器读程序CANRead()、写程序CANWrite()、初始化程序CANInit()、发送程序txdsusb()、接收程序rxdsusb()程序组成。之所以要编写单独的SJA1000的寄存器读、写子程序，这是由SX52芯片只有I/O端口决定的。

选用CAN2.0A协议构建CAN总线控制网络，对SJA1000的初始化主要完成控制寄存器CR、验收代码寄存器ACR、验收屏蔽寄存器AMR、总线定时寄存器BTR0,1和输出控制寄存器OCR的设置。初始化完成后，由总调度模块监控SJA1000控制器。当CAN总线上有数据到达时，它调用接收子程序rxdsusb()，把这一帧数据包存入数据缓冲区B中，然后释放接收缓冲器。同样，当有按CAN2.0A协议格式组合成一帧数据报文在数据缓冲区A中要发送到CAN总线上时，总调度模块将调CAN发送子程序txdsusb()发送。

3.1.2 以太网控制器协议转换模块

以太网控制器协议转换模块主要负责从UDP数据包中解析出完整CAN协议报文，存入数据缓冲区A。同时，可能将数据缓冲区B中的完整CAN协议报文封装成UDP数据报，然后将其发送到以太网上。

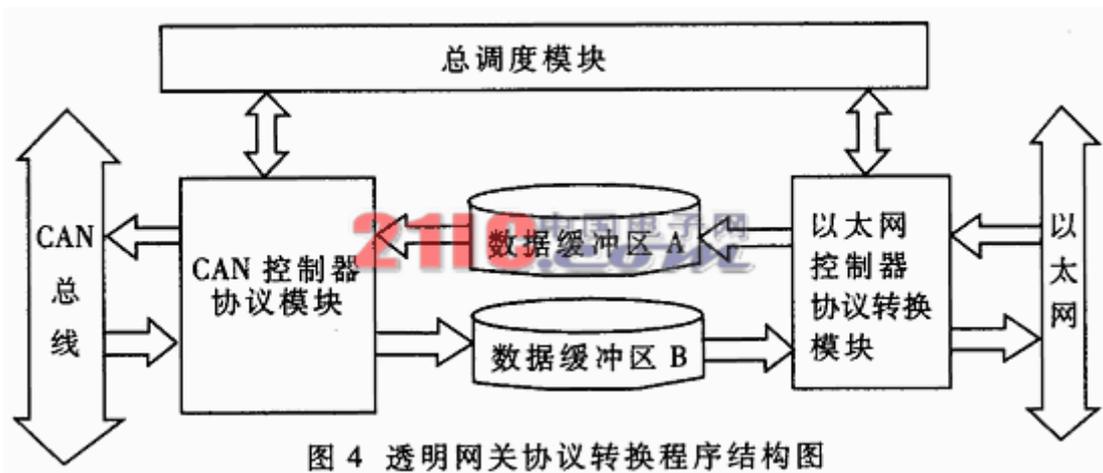


图4 透明网关协议转换程序结构图

在通信传输层采用UDP协议是考虑到CAN协议数据报为短帧形式（每个数据帧最多为8字节）。如果采用TCP传输协议，要传输8字节CAN协议数据，要先通过3次握手建立连接，再传输数据，之后还要通过握手释放连接。这样传输效率对有限的网络资源来说无疑是一种浪费。而UDP是无连接的传输，可以提高网络传输效率，同时，也减轻网关的处理任务。当然，UDP传输协议是不可靠的，对于控制网络来说，是不允许的。为了提高通信的可靠性，采用了回传校验机制。通过实验测试表明这种方式是行之有效的。

以太网控制器协议转换模块主要由以太网卡驱动、ARP、UDP协议的若干个API函数组成，如NICInit()、NICDMAInit()、NICInitTxFrame()、NICSendTxFrame()、NICReadAgain()、ARPCheckIfIs()、ARPSendResponse()、ARPSendStPacket()、ICMPProcPktIn()、UDPApPInit()、IPGenChecksum()、UDPApPProcPktIn()、UDPStartPktOut()和UDPEndPktOut()等。所使用的变量有：remoteIP[3:0]、myIP[3:0]、UDPRxSrcPortMSB、UDPRxSrcPortLS

B、UDPRxDatLenMSB、UDPRxDatLenLSB、UDPTxSrcPortMSB、UDPTxSrcPortLSB、UDPTxDestPortMSB、UDPTxDestPortLSB、DPTxDatLenMSB、UDPTxDatLenLSB等。

系统首次执行或复位时，以太网控制器协议转换模块将首先调用NICInit()和UDPAppInit()等进行NIC、ARP、IP、UDP和应用程序的初始化。初始化完成后，即进入主循环。在主循环中，SX52将反复检测RTL8019AS是否接收以太网帧。当有数据被接收时，SX52调用NICDMAInit()和NICReadAgain()读入以太网帧首部 再调用ARPCheckIfIs()判断接收帧是否为ARP数据。若是ARP，则转入ARPSendResponse()和ARPSendStPacket()子程序进行ARP处理并发送响应ARP数据报；若不是ARP，则判断是否为IP数据报。若非IP数据报则清除该以太网帧；当所接收帧包含IP数据报时，则需进一步判断是ICMP数据报还是UDP数据报文。若是ICMP数据报则执行ICMPProcPktIn()子程序处理ICMP数据报并重发IP数据报；若数据为UDP数据报文，则调用UDPProcPktIn()子程序。该程序将读入UDP数据报文首部的数据并进行相应处理，还原出完整的CAN协议数据报文存入数据缓冲区B中，再通知总调度程序，由总调度程序调用CAN总线控制子程序将CAN协议数据报文发往CAN总线。

反过来，当总调度程序通知以太网控制器协议转换模块将数据缓冲区B中准备好的CAN协议数据发送到以太网上时，它将调用NICInitTxFrame()、UDPStartPktOut()、IPGenCheckSum()、IPStartPktOut()、NICSendTxFrame()、UDPEndPktOut()等子函数进行发送处理，从而实现CAN总线到以太网的数据传输。

3.2 以太网层应用程序设计

以太网上的通信协议一般采用TCP/IP协议。本文采用流行的SOCKET套接字编程，传输层协议选择UDP（用户数据报协议），通过Visual C++编写用户层程序。

WinSock提供了对UDP的支持，通过UDP协议可以向指定IP地址的透明网关发送CAN协议数据，同时也可以通过它接收CAN协议数据。发送和接收方处于相同的地位没有主次之分。利用CAsyncSocket类操纵无连接的数据发送较简单。首先生成一个本地套接口（需要指明SOCK_DGRAM标记）；然后利用int CAsyncSocket SendTo(const void lpBuf int nBufLen UINT nHostPort LPCTSTR lpszHostAddress = NULL int nFlags = 0 发送数据，int CAsyncSocket ReceiveFrom(void lpBuf int nBufLen CString& rSocketAddress UINT& rSocketPort int nFlags = 0 接收数据。利用UDP协议可以使管理主机和SX52网关实现双向的数据通信。同时，这种传输方式也易于使数据SX52网关透明化。

本文介绍了一种低成本、高可靠性、快捷的CAN总线与以太网互连方案。该互连方案保证管理监控层与生产测控层之间的连接，方便了上下层信息交流，满足工厂、变电站等工业场合的应用要求。