

CAN 总线与以太网互连系统^①

邬思奇¹, 戴瑜兴¹, 肖彬², 王卫国³

¹(湖南大学 电气与信息工程学院, 长沙 410082)

²(湖南科技大学, 湘潭 411201)

³(浙江天正电气股份有限公司, 温州 325604)

摘要: 为了实现以太网与 CAN 总线的的数据交换, 构建了一个基于嵌入式的 CAN 总线与以太网互连系统。系统硬件平台以 LPC2119 ARM7 为核心, 包括由 ENC28J60 构成的以太网接口模块、CAN 接口模块、HMI 接口模块; 系统软件平台使用了实时操作系统 μ C/OS-II 和嵌入了 LwIP 协议栈, 并用 VC++ 设计了上位机人机界面。实验调试结果表明, 该系统在智能断路器上运行稳定可靠, 具有一定的实用性和推广价值。

关键词: CAN 总线; 以太网; μ C/OS-II; LwIP 协议栈; ENC28J60

CAN Bus and Ethernet Interconnect System

WU Si-Qi¹, DAI Yu-Xing¹, XIAO Bin², WANG Wei-Guo³

¹(Institute of Electrical and Information Engineering, Hunan University, Changsha 410082, China)

²(Hunan University of Science and Technology, Xiangtan 411201, China)

³(Zhejiang Tengen Electric shares, Wenzhou 325604, China)

Abstract: In order to realize Ethernet and CAN bus's data conversion, Constructed a interconnect system based on embedded technology between CAN bus and Ethernet. Hardware platform of the system let LPC2119 ARM7 as the core, including Ethernet interface module based on ENC28J60, CAN interface module and HMI interface module; Software platform used real-time operating system μ C/OS-II and embedded LwIP protocol stack. The PC user interface was designed by VC++. Experimental debugging results showed, stable and reliable operation of system on voltage circuit breaker, will have certain practicability and commercial worth.

Keywords: CAN bus; ethernet; μ C/OS-II; LwIP protocol stack; ENC28J60

1 引言

CAN 总线由于可靠性高、成本低、应用灵活等诸多优点^[1], 已广泛应用于智能通信网络中, 但收发器驱动能力的限制^[2]使它不适合远距离数据传输及远程控制。以太网技术成熟、通信速度快、软硬件产品丰富和外围技术支持全面^[3], 可以利用网络进行远距离通信, 但在工业控制中仍然存在许多问题尚未解决^[4]。针对目前存在的 CAN 总线网络加大通信距离造成通信速度下降, 干扰大, 危险系数高等缺点的情况^[3], 结合 CAN 总线与以太网自身的优缺点, 本文设计了

CAN 总线与以太网互连系统, 灵活的扩展原有的网络系统, 解决了目前市场上的 CAN 网关用传统以太网控制器所带来的硬件复杂, 占空比高, 造价昂贵等问题, 在成本、功耗、体积、灵活性等方面有了明显的优势。由于系统移植了实时操作系统 μ C/OS-II, 这改善了不能满足工业现场的可靠性和实时性要求, 特别是在连接请求较多或控制任务较复杂时实时性较差的现状。

2 整体系统设计

CAN 总线把单个分散、数字化、智能化的测量和

① 基金项目:浙江省科学技术厅第二批重大科技专项项目(2007C11072)

收稿时间:2010-06-18;收到修改稿时间:2010-07-19

控制设备作为网络节点^[4]，构建成具有自动控制功能的网络控制系统。从程序存储空间、数据存储空间、软件实现难易程度、运行速率、性价比、系统可靠性等因素考虑，系统硬件平台以 ARM7 内核 LPC2119 为核心，其外围有以太网接口电路、CAN 接口电路、HMI 接口电路，结构框图如图 1 所示。LPC2119 内部集成了 2 个 SPI 接口和 2 路 CAN 接口，系统协议转换模块只需外加 CAN 收发器和以太网控制器及其外围电路，便可实现 CAN 与以太网数据通讯，减少了分立元件的数量，增强了系统硬件的稳定性和可靠性。

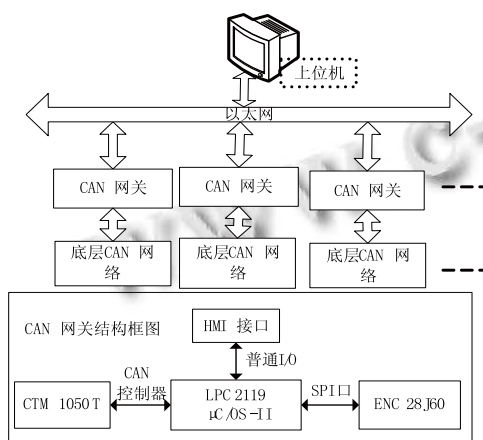


图 1 系统结构框图

系统工作原理：设置 LPC2119 的 SPI 模块为主机，ENC28J60 的 SPI 模块为从机，LPC2119 通过 SPI 口发出命令控制 ENC28J60 的所有操作。上位机采用主从方式，每隔一段时间发送指令与 CAN 底层设备通讯。当 ENC28J60 检测到以太网上有与自身 IP 匹配的上位机数据时，经过一系列的以太网帧格式的认可与响应，抽取出的数据优先存储在内部数据缓冲区，再利用其自带的 CAN 控制器把它发送给底层 CAN 设备；当检测到 CAN 总线上有数据，利用 LPC2119 的 CAN 控制器从总线上提取数据并存储在内部数据缓冲器，根据需要选择合适的以太网协议为数据加上协议头后，数据包通过 SPI 口由 ENC28J60 发往上位机。

3 CAN总线与以太网互连系统的硬件设计

3.1 以太网接口模块的设计

选用 ENC28J60 作为以太网控制器，可通过编程

来解决填充和 CRC 生成、在发生冲突时自动重发、自动拒绝错误数据包^[5]。接口框图如图 2 所示。由于 LPC2119 和 ENC28J60 是主从关系，CLKOUT 输出理论上能为 LPC2119 提供时钟频率，但实际系统运行时，LPC2119 时钟频率并不稳定。ENC28J60 的 SPI 引脚 SO、SI、SCK、CS 分别接 LPC2119 的 MISO0，MISI0，SCK0，P0.8；P0.8 为低电平时，SCK 上升沿数据通过 SI 移入 ENC28J60，SCK 下降沿数据通过 SO 移出 ENC28J60，P0.8 拉为高电平结束操作^[5]；中断引脚 WOL 和 INT 与 LPC2119 的外部中断 0 P0.1 和外部中断 1 P0.3 相连；2 个专用引脚 LEDA、LEDB 接 V3.3，采用灌电流驱动指示网络活动的状态，编程 LED 控制寄存器控制 LED 显示状态^[5]。

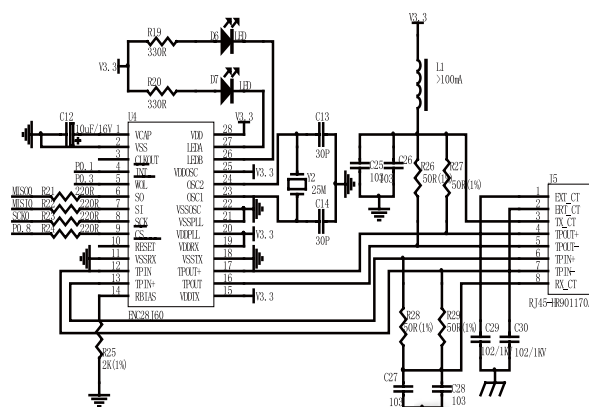


图 2 以太网控制器接口框图

3.2 CAN 接口模块的设计

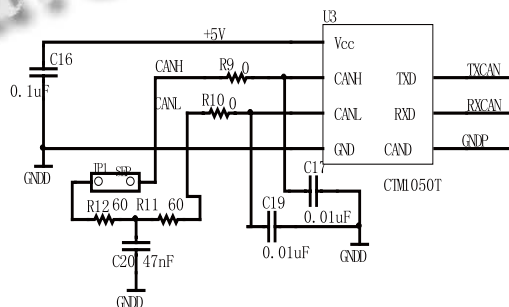


图 3 CAN 接口框图

CAN 总线的通讯速率高达 1Mbps^[2]，常用在环境恶劣的工业场所，对 EMC/EMI 要求很严格^[6]，因此选用集电气隔离、电源隔离、CAN 总线保护、CAN 收

发于一体的 CAN 收发器 CTM1050T。与 LPC2119 的连接图如图 3 所示, CAN 控制器通道 1 的 RD1 与 TD1 通过 CAN 收发器与总线相连。

CANH 和 CANL 的接地电容 C_{17} 和 C_{19} 与噪声源阻抗 R_f 形成了 RC 低通滤波器, 可提高抗电磁干扰性^[6]。由式 (1)可知, C_{17} 越大低通滤波器截止频率 $f_{截止}$ 越低, 抗电磁干扰性越好。

$$f_{截止} = \frac{1}{2\pi R_f C_{17}} \quad (1)$$

当 C_{17} 为 $0.1\mu f$, CAN 波特率 $f = 10kbps$ 时 CAN 数据正常收发, 而 $f = 20kbps$ 时调试界面收不到 CAN 数据。分析可知, 输出阻抗 $R_{输出}$ 和 C_{17} 也构成低通滤波器^[6], 如果截止频率比 CAN 总线波特率低, CAN 信号被当成干扰源滤除了。由此可得 CAN 波特率 $f = 20kbps$ 接近临界状态, 由式 (2) 得 $R_{输出} = 20\Omega$ 。

$$R_{输出} = \frac{1}{2\pi f C_{17}} \quad (2)$$

当 $f = 500kbps$ 时, 由式 (2) 可知 C_{17} 不应超过 $0.016\mu f$ 。该系统中取 C_{17} 和 C_{19} 为 $0.01\mu f$, 经测试正常收发 CAN 数据。

3.3 HMI 接口模块的设计

HMI 接口模块包括 LED 显示和编码开关。LED 用来显示系统协议转换时的工作状态, 编码开关则用来修改 CAN 的波特率、CAN 节点地址和目标 IP 地址。

4 CAN总线与以太网互连系统的软件设计

4.1 $\mu C/OS-II$ 的引入与实现

由于该互连系统对数据的实时性要求比较高, 因此采用实时操作系统 $\mu C/OS-II$ 。根据系统各个任务的重要性与实时性, 把系统软件分成 8 个不同优先级的任务, 如表 1 所示。系统上电复位首先要对系统硬件 LPC2119 和 ENC28J60 进行初始化配置, 以确保系统能够正常工作, 优先级别最高; LED 显示和编码开关处理对实时性要求很低, 重要性也比其它任务低, 因此优先级很低。CAN 与 TCP/IP 转换包括两部分, CAN-TCP/IP 和 CAN-TCP/IP。除此之外, 系统还有三个中断服务程序: 提供周期性信号源的时钟节拍中断^[2]; 把接收到的数据存入内部数据缓冲区的 CAN 接收中断; 处理暂停帧的 CAN 发送中断。

表 1 任务划分表

任务号	任务内容	实时性	优先级
1	系统初始化	最高	6
2	监控系统	很高	7
3	CAN 与 TCP/IP 转换	较高	8
4	ENC28J60 接收数据	高	9
5	ENC28J60 发送数据	一般	10
6	编码开关处理	低	11
7	LED 显示	较低	12
8	空闲任务	最低	63

4.2 TCP/IP 协议的实现

LwIP 协议栈在 TCP/IP 协议实现的基础上减少了对 RAM 的使用^[7], 因此在嵌入式系统中应用比较广泛。LwIP 协议栈实现 ARP、ICMP、IP、TCP、UDP 等协议, 大大缩短了开发周期。操作系统封装层 (sys_arch) 为底层操作系统和 LwIP 协议栈提供了网络接口^[8], 在网络接口中运用不同的结构体 netif, 实现对网络接口的初始化、数据收发以及中断异常处理^[8]。

4.3 CAN 与 TCP/IP 协议转换

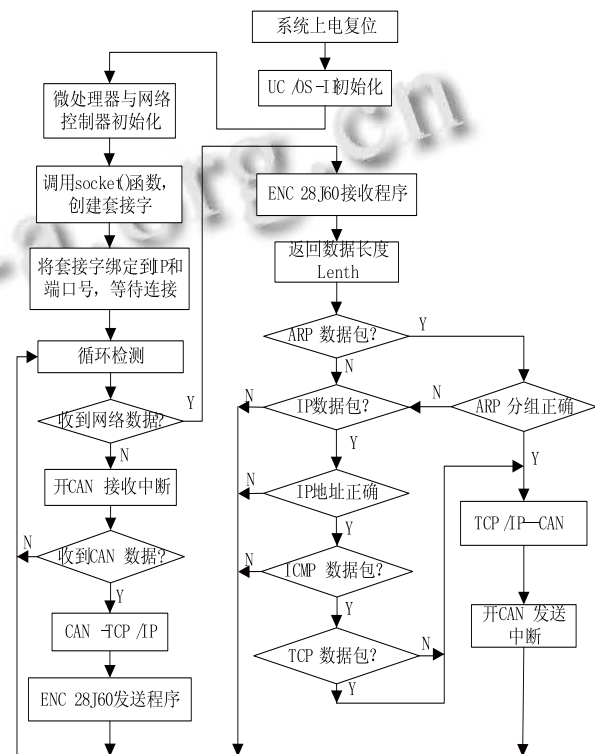


图 4 主程序流程图

系统在上电，完成对 $\mu C/OS-II$ 初始化后，也初始化 LPC2119 和 ENC28J60 相关寄存器，使得整个系统处于就绪状态。由于 LwIP 协议栈实现了 TCP/IP 协议，与 VC++ 开发网络通讯程序相似，采用 Socket 套接字在网络上发送数据和接收数据。系统完成四种数据处理：一是 CAN 和以太网数据的转换；二是 CAN 数据的接收发送；三是以太网数据的接收与发送；四是以太网帧格式的识别与处理。主程序流程图如图 4 所示。

以太网数据处理过程：首先判断网络上是否有数据，如果有，调用 ENC28J60 接收任务，把数据从接收缓冲区读出，并返回收到的数据字节数 Lenth；判断属于哪种网络数据包；调用 TCP/IP-CAN 任务，去掉 IP 头和 TCP 或 UDP 头等，把数据按 CAN 报文格式封装好后存入 CAN 控制器的发送缓冲区；开 CAN 发送中断。如果 Lenth 大于 8，CAN 数据分批发送，否则直接发送。ENC28J60 发送接收数据流程图如图 5 所示。

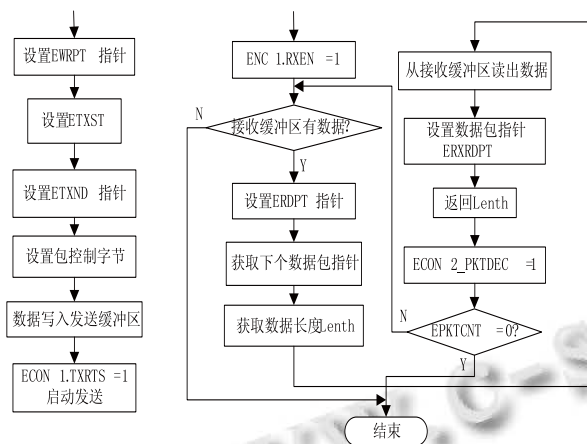


图 5 ENC28J60 发送接收数据流程图

CAN 数据包处理过程：首先检查 CAN 总线上是否有数据，采用中断方式接收 CAN 数据包；调用 CAN-TCP/IP 任务，先对 CAN 报文进行解包，然后在 CAN 数据上添加 TCP 或 UDP 头和 IP 头等，封装成以太网数据包；调用 ENC28J60 发送任务，把数据发往以太网。CAN 发送接收数据流程图如图 6 所示。

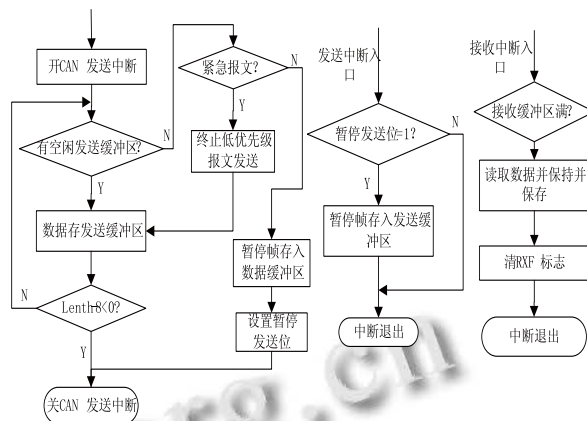


图 6 CAN 发送接收数据流程图

5 上位机设计

上位机程序采用 VC++ 编写。操作人员可以对系统参数进行配置，包括目标 IP 与端口号、CAN 波特率和 CAN 节点地址。应用于智能断路器的上位机截图如图 7 所示，VIEW 类分割成两部分，控制面板和信息面板。上位机采用主从方式，定时查询断路器的实时电流值、壳架等级值、开关状态、工作状态等。图中曲线是 CAN 波特率 500kbps，CAN 节点地址 18，目标 IP 192.168.68.69，端口号为 4000，断路器三相电流的波形图。在控制面板中操作人员可发“分断”控制命令对断路器实现遥控；操作人员也可在菜单项“参数整定”中查询长延时电流、短延时电流、瞬动电流接地故障电流、长延时时间、短延时电流等参数，并根据具体情况进行遥调。

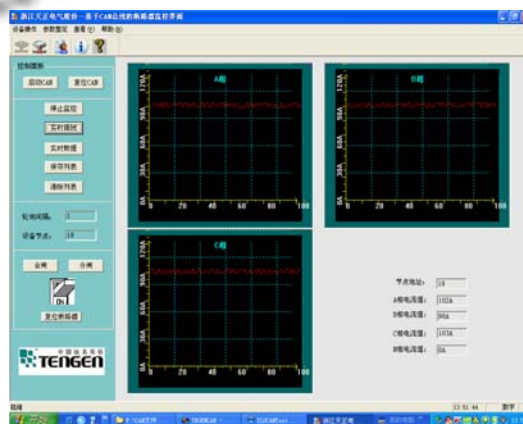


图 7 上位机截图

(下转第 124 页)

务逻辑完全解耦。业务逻辑层使用的是 Spring2.5, 通过 Spring 的 IoC 技术和事务管理机制, 在表现层和数据持久层之间, 起到了很好的衔接作用。数据持久层则采用了 Hibernate3.0, Hibernate 对持久层进行了良好的封装, 降低了程序的复杂度。系统应用实图如图 3 所示。

从精益生产管理系统上线运行至今, 程序运行稳定, 系统响应良好, 同时, 对于该企业进一步有效地实施精益生产, 起到了较大地促进作用。

4 结束语

本文提出了一种新的、基于 AOM+Spring+Hibernate 架构的 Java EE 解决方案, 并详细的阐述了这三种框架技术的整合过程。基于该解决方案, 成功的开发了精益生产管理系统。实践表明, 采用 ASH 架构进行 Java EE 企业级应用开发, 极大地缩短了项目开发周期, 提高了系统的可维护性和可扩展性。同时, 也降低了企业的成本投入。

(上接第 21 页)

6 结语

本文提出了一种 CAN 总线与以太网互连系统的方法。该系统硬件采用了 ARM7 内核 LPC2119 和新型以太网控制器 ENC28J60, 提高系统的稳定性同时节约了成本。同时也提供了一个友好的 HMI, 可以快速方便地修改 IP 地址和 CAN 波特率、掌握系统的工作状态。系统在 LPC2119 和 ENC28J60 硬件平台上移植 μ C/OS-II 和 LwIP 协议栈, 实现了 CAN 网络现场设备与以太网的通讯, 在提高通信速率的同时保证了高可靠性、低通信延时。系统利用 VC++ 开发了一个应用在智能断路器上的上位机监控界面, 对研制我国第四代智能化可通信低压电器提供了便利。可以展望, 该系统将具有可观的商用价值和广阔的应用前景。

参考文献

- 徐喆, 张卓, 闫士珍. 基于 μ C/OS-II 的 CANopen 从节点的实现. 计算机系统应用, 2008, 17(7): 113-118.

参考文献

- Johnson R, Hoeller J. Expert One-on-One J2EE Development without EJB. Version 1.0. Wiley Publishing Inc, 2004: 52-60.
- 林信良. Spring 技术手册. 北京: 电子工业出版社, 2006. 1-10.
- 蔡雪焘. Hibernate 开发及整合应用大全. 北京: 清华大学出版社, 2006. 30-81.
- 深圳市金蝶中间件有限公司. AOM2.3 reference [2010-05-01]. AOM2.3 reference website: <http://infocenter.apusic.com/help/index.jsptopic=/operamasks-sdk/v2.3/index.html>, 2009.
- 韦云东. 机加车间准时制生产监控系统的研制与实现[硕士学位论文]. 重庆: 重庆大学, 2008.
- 朱际华, 孙国强. Wicket-Spring-Hibernate 架构及应用. 计算机应用, 2008, 28(2): 132-134.
- Johnson R. J2EE Development Frameworks. Computer, 2005, 38(1): 107-110.
- 冯润民. 基于 SSH 的高校学生管理系统设计与实现. 计算机工程, 2009, 35(6): 280-282.

- 沈跃, 王家鼎. 基于 ARM 技术的 CAN 总线和以太网的互联设计. 电子技术应用, 2006, 32(5): 56-59.
- Szabo S, Oplustil V. Distributed CAN based Control system for Robotie and Airborned Applications. IEEE on Control, Automation, Robotics and Vision, 2002, 3(11): 1233-1238.
- Li DZ, Wang CA. The implementation of CAN-Ethernet communication system on the missile simulation and detection platform. IEEE 9th International Conference, Kunming, 2008, 11.
- Microchip Technology Inc. ENC28J60 Stand Alone Ethernet Controller with SPI Interface. [2008-2-13]. <http://www.microchip.com>.
- Semiconductors P. Data Sheet TJA1050 High Speed CAN transceiver. [2000-5-10]. <http://www.zlmcu.com>
- Dunkels A. Design and Implementation of the LwIP TCP/IP Stack. Swedish Institute of Computer Science, 2001, 5(2): 16-22.
- Thomas F. An Introduction to TCP/IP for Embedded Engineers. Embedded Systems, 2002, 5(3): 23-28.