

# 基于 MPC8260和 VxWorks实现快速以太网通信

顾永红,杨巧丽

(总参第 63研究所,江苏省南京市 210007)

**摘要:** MPC8260芯片因其强大的通信处理能力和多种协议的支持而被广泛应用于通信和网络领域。以 MPC8260微处理器为例,讨论在嵌入式系统中实现快速以太网通信的硬件设计方法,分析了 VxWorks网络系统的结构,描述了基于 VxWorks的以太网驱动程序的开发方法及其流程。对所有基于 MUX(多路选择)机制下的 VxWorks网络设备驱动程序开发者具有参考价值。

**关键词:** VxWorks;以太网;嵌入式系统;增强型网络驱动

**中图分类号:** TP393.11

## 0 引言

摩托罗拉 MPC8260微处理器芯片因其强大的通信处理能力和多种协议的支持而被广泛应用于通信和网络领域。本文以 MPC8260为例,探讨在嵌入式系统中实现基于 VxWorks操作系统的快速以太网通信的方法。

## 1 快速以太网通信接口的硬件实现

MPC8260的 3个 FCC(快速通信控制器)均可通过 MII(媒体独立接口)支持快速以太网,实现以太网中的 MAC(媒体访问控制)层功能。在设计中选用 FCC1接口作为以太网接口。以太网 PHY(物理层)收发器选用 Intel公司的 LXT971A,变压器选用中山汉仁公司的 HR601680,采用 RJ45标准物理接口。以太网口硬件设计如图 1所示。

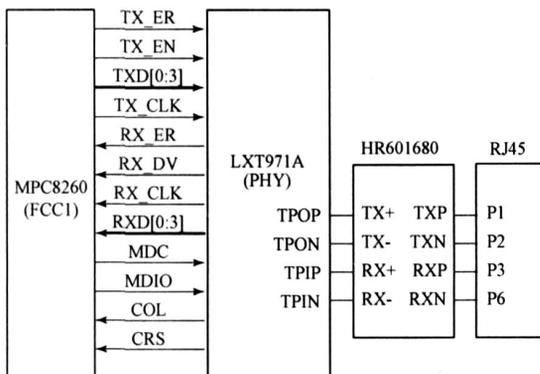


图 1 以太网口硬件设计简图

LXT971A 支持 10BASE-TX、100BASE-TX/FX 等物理连接方式,拥有半双工、全双工、自适应等多种工作模式,拥有内嵌的 LED(发光二极管)驱动显示网口

的工作状态,采用通用的 3.3V 供电。在本设计中, LXT971A 通过 MII 与 MPC8260 的 FCC1 连接, MPC8260通过 LXT971A 的串行控制管理(MDD(管理数据输入输出)和 MDC(管理数据时钟))接口对 LXT971A 内部寄存器进行配置,实现工作模式的选择。但由于 LXT971A 是完全自适应的,因此,在实际的工作中不需要对 LXT971A 进行任何配置。需要注意的是, MPC8260的引脚存在复用的问题,因此,软件必须将所涉及的引脚设为对应的 FCC1通信口的功能。

## 2 快速以太网驱动的设计与实现

### 2.1 VxWorks网络系统结构

VxWorks支持两种形式的网络驱动:一种是 BSD(Berkeley Software Distribution)驱动;另一种是 END(增强型网络驱动)是 VxWorks所独有的,其主要特点是增加了一个 MUX模块来管理 END设备。目前比较通用的是采用 END网络驱动程序。在本系统的网络驱动中选用 END驱动作为以太网驱动方式,其网络系统结构如图 2所示。



图 2 VxWorks网络系统结构

收发数据时,网络协议层通过套接字(Socket)获取应用层提供的信息,经由 MUX(多路选择)层

选择 END 设备。MUX 层将网络协议与网络设备的驱动区隔开来,使得网络协议不依赖于具体的硬件接口,而与硬件有关的代码都封装在 BSP 中,底层由 END 提供驱动。因此,在基于 VxWorks 的以太网通信实现的设计中,主要的工作是进行底层网络设备(硬件)设计和 END 驱动程序的开发。

## 2.2 END 驱动的实现

以太网在 VxWorks 下作为 END 设备,由通信处理器模块通过 FCC 接口进行管理。END 驱动的工作过程一般分为 3 个步骤完成,即初始化 END 设备、装载 END 设备以及启动 END 设备。风河公司提供了基于大多数嵌入式微处理器的评估板的 BSP 参考源程序,根据实际方案修改其中与网络通信相关的部分程序,即可实现 END 驱动在实际设计中的应用。

### 2.2.1 END 设备的初始化步骤

为了便于理解网络设备程序的装载过程,首先分析图 3 所示的在 VxWorks 映像启动时的网络初始化顺序。

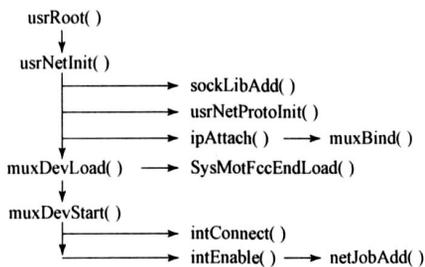


图 3 网络初始化步骤

系统启动后, VxWorks 首先执行 usrRoot() 函数, 安装库程序和创建设备。接着 usrRoot() 调用 usrNetInit() 函数完成网络的初始化, 包括通过 muxDevLoad() 调用 SysMotFccEndLoad() 装载设备表 endDevTbl[] 中描述的设备以及调用 ipAttach() 完成网络协议的安装等。通过这些过程完成网络装载并使其处于准备接收或发送数据状态。设备装载完成后, muxDevLoad() 调用 muxDevStart() 启动函数, 该函数通过 intConnect() 和 intEnable() 实现中断服务的注册并打开中断宏。当有数据交换时, 打开中断服务程序, 调用排列网络任务处理函数 netJobAdd(), 指派网络系统任务 NetTask, 由 NetTask 处理数据接收和发送任务。

### 2.2.2 相关 BSP 的配置

在 VxWorks 中, 网络设备驱动程序装载时, 首先

要通过对所处 BSP 中文件进行相应的配置才能实现。

#### 1) 接口引脚的定义和使能

以太网是由 MPC8260 的以太网控制器通过连接一个外部 PHY 芯片实现的, 通过对 MPC8260 的可编程并行口进行寄存器设置, 定义这些引脚为以太网接口信号。这部分功能在 sysLib.c 中实现。

另外, 需要同时配置读写以太网 PHY 芯片内部寄存器的信号, 这可以通过修改 sysFccEnetAddrGet, sysFccMiiWr, sysFccMiiRd 等子过程来实现, 详细的配置数据可以参照 MPC8260 并行口寄存器的配置表, 这里不一赘述。

#### 2) 以太网物理端口的定义

针对以太网接口, 定义使用的 FCC 号、TBD 和 RBD 数目、PHY 物理地址和使用的通信速率操作模式以及用户标识等。这部分功能在 sysMotFccEnd.c 中实现。

```

#define MOT_FCC_NUM 0x1 //使用 FCC1
#define MOT_FCC_TBD_NUM 0x40 //TBD
#define MOT_FCC_RBD_NUM 0x20 //RBD
#define MOT_FCC_PHY_ADDR0 0x00 //PHY地址
#define MOT_FCC_DEF_PHY_MODE PHY_100BASE_TX
//使用 100BASE_TX 模式
  
```

#### 3) 添加驱动的调用入口

在 configNet.h 中定义 END 设备驱动程序入口表 END\_TBL\_ENTRY 结构的数组 endDevTbl[], 将驱动装载函数 SysMotFccEndLoad 的入口点及相关参数添加到 endDevTbl[]。

## 3 结束语

本文介绍了在 MPC8260 为核心的嵌入式系统中, 基于 VxWorks 操作系统的快速以太网通信的硬件设计方法和驱动开发过程, 该设计已在实际工作中顺利实现。

## 参考文献

- [1] 孔祥营, 柏桂枝. 嵌入式实时操作系统 VxWorks 及其开发环境 Tomado[M]. 北京: 中国电力出版社, 2002.
- [2] 周启平, 张杨. VxWorks 下设备驱动程序及 BSP 开发指南[M]. 北京: 中国电力出版社, 2004.

顾永红 (1966-), 女, 高级工程师, 主要从事卫星通信与嵌入式系统开发工作。

(下转第 72 页)

## 参 考 文 献

- [1] 徐涛, 闫科, 赵景林, 等. 基于 ModBus协议的串行接口实现与 DCS通信 [J]. 工业控制计算机, 2005, 15 (3): 56-57.
- [2] Modicon Inc. Modicon modbus protocol reference guide [EB/OL]. [http://www.modbus.org/docs/PI\\_MBUS\\_300\\_1996](http://www.modbus.org/docs/PI_MBUS_300_1996)
- [3] Schneider Electric. MODBUS over serial line specification &

implementation guide V1. 0 [EB/OL]. <http://www.Modbus.org>, 2002

- [4] 马忠梅, 籍顺心, 张凯, 等. 单片机的 C语言应用程序设计 [M]. 北京: 北京航空航天大学出版社, 2000.
- 陈 雄 (1983-), 男, 硕士研究生, 主要研究方向为机电控制。

## Design of Frequency Conversion Control System Based on ModBus Communication Protocol

CHEN Xiong, FANG Jianglong, TAO Yin

(College of Mechanical Engineering, Shanghai University of Science and Technology, Shanghai 200093, China)

**Abstract:** Frequency conversion control system works mostly in a high voltage, high temperature environment, the stability and anti-interference must be highly needed. The ModBus protocol has reliable data transmission, rapid response and strong anti-interference capability. How to design the frequency conversion control system based on ModBus protocol RTU transmission mode is introduced. Serial communication uses inquiry to exchange the data between the master and slave, whose connecting method is also presented. The anti-interference ability and stability of the frequency converter are enhanced.

**Keywords:** ModBus RTU; frequency conversion control system; frequency conversion; RS-485

(上接第 65 页)

## Research on Tri-channel-based MAC Scheme in Ad hoc

SUN Wang

(Institute of Signal Processing and Transmission, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract:** It presents a Tri-channel-based MAC scheme in Ad Hoc networks to resolve the hidden & exposed terminal problem. It enables the node in hidden or exposed state to work, so that the node could access channel in good time correspondingly. It avoids the collision that result in resending of data packets by transmitting ACK in special channel. The combination of frequency-division-multiplexer and time-division-multiplexer scheme enables the tri-channel to allow dual-link for data-transmitting.

**Keywords:** Ad hoc network; hidden & exposed terminal; tri-channel-based; MAC

(上接第 67 页)

## Implementation of Fast Ethernet Communication Based on MPC8260 and VxWorks

GU Yonghong, YANG Qiaoli

(63rd Research Institute of PLA General Staff Headquarters, Nanjing 210007, China)

**Abstract:** MPC 8 2 6 0 has been widely used in communication and network areas because of its powerful ability of communication processing and support to many protocols. Taken MPC8260 as an example, the hardware design of the fast Ethernet in embedded system is discussed and the architecture of network system in VxWorks is analyzed. Then the driver programming of Ethernet based on VxWorks is described in detail. It has a reference value for other network device driver developers.

**Keywords:** VxWorks; Ethernet; embedded system; END (enhanced network driver)