

嵌入式 uIP TCP/IP 协议栈在基于 DSP 系统中的应用

谢文娣

(安徽新华学院, 安徽 合肥 230088)

摘要: 搭建了一个基于 DSP 嵌入式系统的以太网接入硬件平台, 给出了软件设计思路及具体实现方法, 对开源的 uIP 协议栈进行裁剪和修改, 完成了设备驱动程序的开发和嵌入式 uIP 协议栈的移植, 最终实现了嵌入式系统接入以太网的功能, 且其实现过程简单、成本较低。

关键词: DSP; uIP 协议栈; 接入以太网

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2013)09-0045-03

Application of embedded uIP TCP/IP protocol stack based on DSP system

Xie Wendi

(Anhui Xinhua University, Hefei 230088, China)

Abstract: In the first place, this paper builds an Ethernet Access of hardware platform based on DSP embedded system. In the next place, it gives the design ideas of software and the specific method in realization, and makes some cutting of the light open-source uIP stack; besides it has completed the development of device driver program and the software transplant of embedded uIP protocol stack. In the end this thesis accomplishes the embedded system accessed to the Ethernet. Its realization is simple and the cost is rather low.

Key words: DSP; uIP protocol stack; Ethernet Access

近年来, 嵌入式技术发展迅速并在许多电子产品(如手持式、便携式)中都有广泛的应用^[1], 而嵌入式产品网络化是当今社会一股不可抵挡的研究潮流。嵌入式系统必须具有网络互连功能才能快速地与设备通信, 以太网是目前使用最广泛的局域网技术。由于有些嵌入式系统要求网络的实时性较高, 因此本文使用精简 TCP/IP 协议栈——uIP 协议栈来实现基于 DSP 的嵌入式系统以太网接入功能, 下面详细阐述其设计与实现。

1 系统硬件平台

TMS320C5000 系列是 16 bit 定点 DSP, 通常被应用在如 VoIP 以及 IP 电话网关等通信领域中。基于 DSP 的嵌入式系统中选择的数字信号处理芯片是 TI 公司的 TMS320VC5510 芯片, 网卡芯片选择的是由 Cirrus Logic 公司生产的 CS8900AS 芯片。系统硬件接口基本结构图如图 1 所示。

TMS320VC5510 芯片是由 TI 公司生产的第一个基于定点 TMS320C55x 核, 具有低功耗、高性能的 16 bit 数字信号处理器^[2]。Cirrus Logic 公司专门设计了以太网控制



图 1 系统硬件平台

器 CS8900A, 这款网卡芯片主要应用在对可靠性要求极高的工业控制领域中^[3]。

2 系统软件设计

软件上, DSP 完成数据包的读和写工作, 主要是通过软件编程来控制 CS8900A 以太网控制芯片中的各寄存器, 从而实现嵌入式系统接入以太网的通信功能^[4]。图 2 所示为 uIP 协议栈与系统底层和应用程序之间的接口关系图。

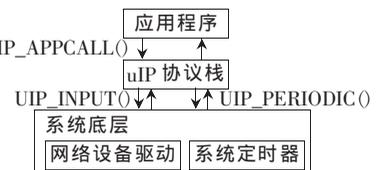


图 2 uIP 协议栈与各层之间的接口关系图

2.1 底层驱动程序的设计

本系统采用查询方式来完成 CS8900A 芯片的软件

网络与通信 Network and Communication

编程控制工作。这一工作主要分为以下 3 个过程：

(1) 首先是完成以太网控制芯片的初始化工作,由初始化函数 `ethernet_driver_init(void)` 来完成发送/接收配置寄存器的相关设置工作,如对 CS8900A 进行软件复位、设置以太网接口参数以及设置以太网的硬件地址。

(2) 若发送数据,则由数据发送函数 `ethernet_driver_write()` 先将发送命令写入 CS8900A 的 TxCMD 寄存器中,其次在 CS8900A 的 Txlength 寄存器中写入将要发送的数据帧长度,接着检查 BusStatus 寄存器的 PPD,待 PPD 处于空闲状态时将要发送的数据帧写入 CS8900A 中,最后 CS8900A 将自动决定数据帧以相应的报文根据 TxCMD 寄存器的设置向以太网上发送。

(3) 若以太网接收到数据包,则由数据接收函数 `ethernet_driver_write()` 读取数据包,CS8900A 将根据接收配置寄存器的设置自动对接收的报文进行预处理。

2.2 uIP 协议栈的设计

由瑞典计算机科学学院的 Adam Dunkels 开发的 uIP 协议栈是一种免费的、可实现的且非常小的 TCP/IP 协议栈,它是为 8 bit 和 16 bit 处理器专门设计的^[5]。uIP 协议栈是使用 C 语言编写的,它移植起来方便而且代码也相当简洁。uIP 协议栈去掉了完整的 TCP/IP 中不常用的功能,简化了通信流程,占用的资源少,处理速度快。

基于 DSP 的嵌入式系统 uIP 协议栈中,根据需要只实现了 ARP、IP、ICMP、UDP 协议。TCP 提供面向连接的、可靠的、点对点的服务^[6],但 TCP 开销大,传输速度慢;虽然 UDP 是面向无连接的,由于 UDP 没有可靠性的保证机制,因此能以较快的速度进行数据通信;UDP 协议的开销很小,传输率比 TCP 高出很多,实时性更强,所以 uIP 协议中采用 UDP 协议作为运输层协议。该 uIP 协议栈的网络体系结构如图 3 所示。

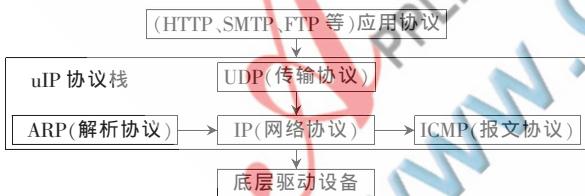


图 3 uIP 协议栈体系结构

2.3 uIP 协议栈的实现

uIP 协议栈的实现主要集中在 `uip.c` 的源文件里,代码简洁。图 4 所示为 uIP 整个系统的主程序流程图。在这个主程序流程图中,初始化工作是由底层硬件完成的,主要是对物理层的一些接口进行配置和绑定。uIP 协议栈接收数据包的过程实际上就是对数据包进行解析,当网络驱动设备发送数据包过来时,首先读取数据包判断是 IP 包还是 ARP 包,如果是 ARP 包,则进行 ARP 处理;若是 IP 包,则判断其上层协议是 UDP 协议还是 ICMP 协议,再分别对其进行处理。当有数据发送时,网络驱动设备的硬件将自行判别数据包并且完成数据包的封装工作。

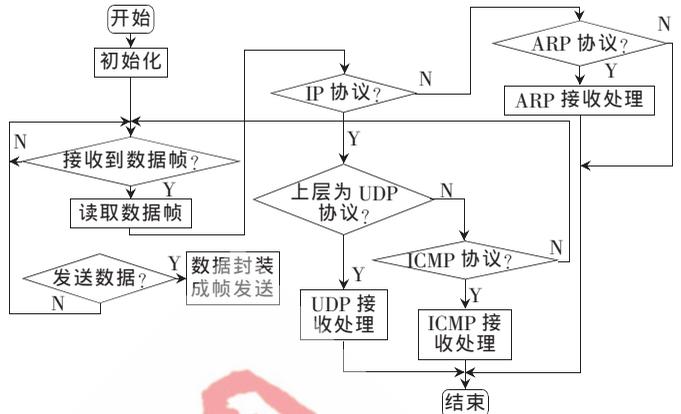


图 4 uIP 协议栈数据包处理流程图

2.3.1 ARP 协议

ARP 协议是由源文件 `Uip_arip.c` 实现的,其中包含 `uip_arip_init()`、`uip_arip_timer()`、`uip_arip_update()`、`uip_arip_aripin()` 和 `uip_arip_out()` 5 个函数。`uip_arip_init()` 完成 ARP 的初始化工作,初始化 ARP 首部、入口地址等。`uip_arip_timer()` 完成动态映射表的老化功能,这个函数每 10 s 被调用一次,若动态映射表里的映射关系在连续 20 min 内没有发生变化,则其映射将失效。`uip_arip_update()` 完成动态映射表的更新功能。`void uip_arip_aripin()` 对接收到的 ARP 分组进行处理。`uip_arip_out()` 完成 ARP 请求的自动发送功能,它是在即将发送一个 IP 分组前被调用。

2.3.2 IP 协议

IP 协议由 `ip_recv` 和 `ip_send` 两个函数段来实现。`ip_send` 函数段完成 IP 数据包的发送功能,接收上层协议传送过来的数据包加上 IP 首部进行封装,然后设置 `srcipaddr`、`destipaddr`、`proto` 及 `ipchksum`,最后交给下层协议处理好发送;`ip_recv` 函数段完成 IP 数据包的接收功能,当收到数据包时首先判断其 `vhl`、`length`、`destipaddr` 及 `ipchksum` 是否正确,然后根据 IP 报文首部的类型码将该数据包交给相应的上层协议处理。

2.3.3 ICMP 协议

ICMP 协议主要由标志段 `icmp_input` 实现。ICMP 协议在实现时比较简单,具体操作是将 ICMP 的报文字段从回送类型变成回送应答类型,再另外计算 `checksum` 并修改 `checksum` 字段。

2.3.4 UDP 协议

UDP 协议主要由函数 `udpfound` 和 `udpsend` 两个标志段实现。`udpfound` 函数段负责处理收到的 UDP 数据,包括取出源 IP 地址、目的地址和源端口,并判断它的端口号和校验和是否都正确,如果正确再获取目的端口,调用应用程序对接收到的数据包进行相应处理,否则丢弃。`udpsend` 函数段完成 UDP 数据包的发送功能,如果接收到来自应用程序的数据,而这时又有数据要发送,那么首先分配空间给待发送的数据,然后拷贝数据到

网络与通信 Network and Communication

UDP 数据包的数据部分,并对 UDP 首部各个字段对应的源端口号和目的端口号进行设置,最后再调用 ip_send 函数将欲发送的数据交给 IP 协议发送。

3 系统仿真

利用计算机进行仿真时,在 uIP 协议栈中,将目标机(也就是开发板)的 IP 地址设置成 129.27.137.103,区别于 PC 的 129.27.137.104,即构成局域网使得它们能在同一个网段内进行通信。假设已经建立了一条成功的通信链路,这时应该出现当服务器发出命令,目标机立即会给出回应这样的结果。DOS 操作系统下 ICMP 测试图如图 5 所示。从图 5 中可以得知,发送的 4 个 ICMP 数据包均得到了回应,其中最长的回应时间为 18 ms,最短的回应时间不到 1 ms,平均回应时间为 10 ms。这时说明开发板与计算机建立了一条可靠的连接,ping 通信成功,为系统各功能的实现,提供了可靠的基础。

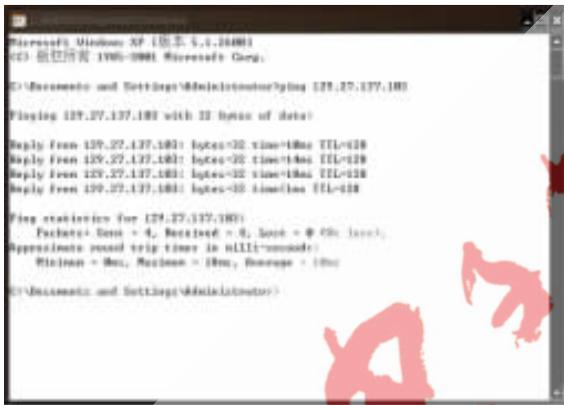


图 5 DOS 操作系统下 ICMP 测试图

本文利用计算机进行了仿真测试,完成了 DSP 与 PC 的 ping 通信,实现了基于 DSP 的嵌入式系统接入以太网的功能。但仍需解决的问题是,由于 UDP 协议不可靠、面向无连接,难免会出现数据包乱序到达、数据包丢失的现象。因此针对这些现象可以对数据进行一些处理,如对每一组数据进行顺序标记,这样可以在应用层发现数据的丢失和乱序,从而予以更正;在数据包前后加一校验码保证接收数据正确等,这将在后续工作中讨论。

参考文献

- [1] 王采莲,郭志强.嵌入式 TCP/IP 协议栈在指挥网络中的实现[C].AECC 专题学术研讨会论文集,2007,20(5):162-163.
- [2] TI.TMS320VC5510AI DataSheet[Z].2004.
- [3] Cirrus Logic. CS8900A Product Data Sheet[EB/OL]. [2007-08-01]http://www.cirrus.com.
- [4] 岳世为,尹为民.uIP 协议栈在基于 DSP 以太网通信系统中的应用[J].计算机与数字工程,2010,38(1):187.
- [5] DUNKELS A. The uIP Embedded TCP/IP Stack[EB/OL]. [2006-06-01]http://www.Sics.se.
- [6] [阿根廷]Scaglia Sergio.嵌入式 Internet TCP/IP 基础、实现及应用[M].潘琢金,徐蕾,拱长青,等,译.北京:北京航空航天大学出版社,2008.

(收稿日期:2013-01-08)

作者简介:

谢文娣,女,1987 年生,硕士研究生,主要研究方向:网络与通信。