

32 位 ARM 微控制器的 IPX 通信模块设计

张海涛, 聂诗良, 唐东明, 韩 宾, 王 昕
(西南科技大学信息工程学院, 绵阳 621010)

摘 要: 以 32 位 ARM 微控制器与 DSPEC 之间的 IPX 通信为例, 介绍了以太网控制器 RTL8019AS 与 ARM 的接口、RTL8019AS 以太网驱动程序设计、IPX 数据帧格式及含义, ARM 实现 IPX 通信数据包的打包与解包过程。实验结果表明, ARM 的 IPX 通信模块与 DSPEC 之间的 IPX 数据传输是高速可靠的, 解决了 ARM 的 IPX 通信问题。

关键词: 32 位 ARM 微控制器; IPX 通信; RTL8019AS; DSPEC

Design of IPX Communication Model for 32 Bits Microcontroller ARM

ZHANG Haitao, NIE Shiliang, TANG Dongming, HAN Bin, WANG Xin

(School of Information Engineering, South West University of Science and Technology, Mianyang 621010)

【Abstract】 This paper considers IPX communication between 32 bits microcontroller ARM and DSPEC as the example, it is introduced detailedly, that the interface between RTL8019AS and ARM, and the design of RTL8019AS ethernet drive program, and data frame format and its meanings of IPX protocol, and packing program and unpacking program design of IPX communication based on ARM. The experiment result shows it has high speed and reliability, that ARM gets the block experiment data from DSPEC by the IPX communication, and the IPX communication is realized by the method for ARM.

【Key words】 32 bits microcontroller ARM; IPX communication; RTL8019AS; DSPEC

网间数据包交换协议(Internet packet exchange, IPX)是一种无连接的数据包传输协议, 具有通信速度快、效率高、路由选择灵活等特点, 可帮助数据包选择路径、交换并寻址, 将它准确地送到目的站点。因此, IPX 通信协议目前在可网络集成的测量设备中得到了广泛应用, 如美国 ORTEC 公司的多道分析仪 (DSPEC) 等。32 位 ARM 微控制器具有计算精度高、速度快、低功耗、存储容量大、可选功能多等优点, 适合用作较复杂任务的数据计算与控制应用系统的中央处理单元。但是, ARM 微控制器及其开发板目前还没有 IPX 通信的功能, 因此, 为了实现 ARM 与具有 IPX 通信协议的测量设备进行 IPX 通信, 本文以 DSPEC 为例, 研究并实现了 ARM 的 IPX 通信模块程序。此外, 采用的 ARM 是 PHILIPS 的 32 位 ARM 微控制器 LPC2214 及其 ARM2200 开发板, ARM2200 开发板上集成了 RTL8019AS 以太网控制器及其 RJ45 连接器; DSPEC 提供了以太网同轴细缆 BNC 接口 (10BASE-2) \ IPX 通信的数据收发命令格式和用户数据格式。

1 ARM 与 DSPEC 之间的硬件接口

ARM 与 DSPEC 之间通过以太网控制器 RTL8019AS 和交换机相连接, 交换机是为了实现 ARM2210 开发板上 RJ45 接口与 DSPEC 的同轴细缆 BNC 接口之间的连接转换, 如图 1 所示。

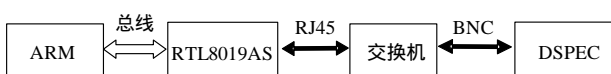


图 1 ARM 与 DSPEC 的硬件接口示意图

RTL8019AS 的功能是在 ARM 与 DSPEC 之间完成数据包与电信号之间的转换。在发送阶段 ARM 将要发送的数据

按照 IPX 数据帧格式打包, 并将数据包按照规定的格式写入到 RTL8019AS 存储空间中并启动发送。RTL8019AS 将该数据包转换成物理帧格式通过物理信道传输给 DSPEC。在接收阶段, RTL8019AS 将接收到的物理信号还原成数据, 存放在芯片存储空间中以供 ARM 读取。

2 IPX 数据帧格式说明

IPX 数据包采用固定结构的帧格式, 在数据帧中包括了数据传输的地址、数据等字段信息, 其具体结构如表 1 所示。

表 1 IPX 数据帧格式

校验和	数据包长度	传输控制位	报文类型	目的地址	源地址	数据
-----	-------	-------	------	------	-----	----

- (1) 校验和: 占 2 个字节, IPX 通信中未用, 固定为 0xffff。
- (2) 数据包长度: 占 2 个字节, 表示整个 IPX 数据包的长度, 包括了 IPX 报头和要传输的数据的长度。
- (3) 传输控制位: 占 1 个字节, 表示传输的报文所经过网络的个数。ARM 与 DSPEC 通信中始终为 0x00。
- (4) 报文类型: 占 1 个字节, 表示不同报文的类型, IPX 通信中为 0x04。
- (5) 目的地址: 占 12 个字节, 包括了目的网络号、目的节点号和目的套接字。目的网络号表示接收站点所在的网络号, ARM 与 DSPEC 的通信中为 0x00000000, 占 4 个字节; 目的节点号与接收站点的物理地址 (接收站点的 MAC 地址) 相对应。如果发送的是广播包, 该节点号为 0xffffffff, 占 6 个字节; 目的套接字, 占两个字节, 在 ARM 与 DSPEC 的通信始终为 0x0455。

基金项目: 中国工程物理研究院资助项目 (20050320532)

作者简介: 张海涛(1981 -), 男, 硕士生, 主研方向: 模式识别与智能系统; 聂诗良, 硕士、副教授; 唐东明、韩 宾、王 昕, 讲师

收稿日期: 2006-04-25 **E-mail:** zht_sir@163.com

(6)源地址:占 12 个字节,包括源网络号、源节点号、源套接字。它们的具体含义与目的地址中目的网络号、目的节点号和目的套接字相同。

(7)数据:要传输给目的站点的可用数据。即要从 DSPEC 读取的实验数据,有 8 192 个十六进制数,每个数据占 4 个字节,共 32KB,每次从 DSPEC 读取 1 024 个字节,共分 32 次读取。

3 ARM 的 IPX 通信模块设计

ARM 与 DSPEC 之间的 IPX 通信模块可分为 IPX 协议栈、以太网层和 RTL8019AS 以太网驱动层 3 个部分,如图 2 所示,往右是发送方向,往左是接收方向。



图 2 IPX 通信程序的 3 个部分

首先是 IPX 协议栈,所要发送的数据通过该协议栈,将数据打包成符合 IPX 数据帧格式的数据包(IPX 数据包);接下来为以太网层,IPX 数据包经过该层程序处理后,在前面加入了以太网报头,使该数据包(以太网层数据包)符合 IEEE802.3 标准;之后是以太网驱动层,以太网数据包经过 RTL8019AS 驱动程序写入到 RTL8019AS 存储区中。启动发送后的工作就由 RTL8019AS 自动完成。RTL8019AS 主要完成数据包到电信号的转换,使信号在物理信道上传输。以上描述的是数据的发送,数据接收是发送的逆过程。

3.1 IPX 协议栈程序设计

IPX 协议栈程序分为发送和接收部分,需要两个接口函数供上层数据发送层和下层以太网层调用,定义了两个函数如下:

```
IPX_Send_Packet(struct Packet *IPX_Send_Data); //IPX 发
//送程序
IPX_Received_Packet(uint8 *IPX_Received_Data); //IPX 接
//收程序
```

其中 IPX 发送程序提供给上层数据发送层调用,数据在通过 IPX 协议栈后在前面加上 IPX 报头,参数 IPX_Send_Data 指针指向数据发送层数据包的地址。IPX 接收程序提供给以太网层调用,用于将接收到的数据进行解包,参数 IPX_Received_Data 为通过以太网层解包后的数据。在接收数据时要根据所接收到的数据包的目的网络地址与本地网络地址(即 ARM 的网络地址)是否一致,来判断是否接收该数据包。如果一致就接收,否则就丢弃。

在程序中按如下结构体定义 IPX 报头,其长度为 30 个字节。

```
struct IPXHEADER
{
    uint16 checksum; //校验和*/
    uint16 length; //IPX 分组长度*/
    uint8 transport_control; //传输控制*/
    uint8 packet_type; //分组类型*/
    uint8 dest_network_number[4]; //目的网络地址*/
    uint8 dest_network_node[6]; //目的节点地址*/
    uint16 dest_network_socket; //目的套接字*/
    uint8 source_network_number[4]; //源网络号*/
    uint8 source_network_node[6]; //源节点地址*/
    uint16 source_network_socket; //源套接字*/
} IPX_Header;
```

在该结构体中,IPX 分组长度 length 等于要发送数据的长度加上 IPX 报头的长度(即 30 个字节)。在发送数据时,目的节点地址等于目标的 MAC 地址(如 DSPEC 的 MAC 地址为 0x00A03C000202,占 6 个字节),如果发送的是广播包

该节点地址为 0xffffffff,共 6 个字节;源节点地址为本地 MAC 地址,可以任意设置,但是必须要与 RTL8019AS 驱动程序中所设定的 MAC 地址相一致。其他参数的设定已在 IPX 数据帧格式中说明了。

另外还定义了一个结构体作为 IPX 协议栈层与以太网层的接口。其中指针 Stptr 指向上一层数据包的地址(即 IPX_Send_Data),Length 为 IPX 报头的长度(即 30 个字节),Send_Data 指向 IPX 报头的地址(即&IPX_Header)。结构体定义如下:

```
struct Packet {
    struct Packet *Stptr;
    uint32 Length;
    uint8 *Send_Data;
} IPX_data;
```

3.2 以太网层程序设计

IPX 通信符合 IEEE802.3 标准,以太网层报头的数据帧格式如表 2 所示。

表 2 以太网数据帧格式

目的 IP 地址(6B)	源 IP 地址(6B)	数据类型/长度(2B)
--------------	-------------	-------------

其中,目的 IP 地址在发送广播包时为 0xffffffff,共 6 个字节,否则为目标物理地址(即 MAC 地址);源 IP 地址也占 6 个字节。数据类型/长度可以表示数据包的类型或者是数据包的长度,占 2 个字节,在 IPX 通信中该段表示数据包的长度,等于整个 IPX 数据包的长度。在数据长度之后是 IPX 数据包。以太网报头如下,长度为 14 个字节。

```
struct Ipphernet {
    uint8 DestMacId[6]; //目的 IP 地址*/
    uint8 SourceMacId[6]; //源 IP 地址*/
    uint16 DtaaLength; //数据长度 LEN*/
} Ethernet_Header;
```

在以太网层程序中设计两个函数接口,函数定义如下:

```
Send_Ethernet_Data(struct Packet *TxdData); //以太网层发送
//程序
Rec_Ethernet_Packed(uint8 *RecData); //以太网层接收
//程序
```

其中以太网发送程序供 IPX 协议栈层调用,作用是在 IPX 数据包中加入以太网报头。参数 TxdData 指向&IPX_data。接收程序供 RTL8019AS 以太网驱动程序调用,将接收到的数据包的目的 IP 地址与本地的源 IP 地址相比较,判断是否接收该数据包。同样在以太网层要定义一个 struct Packet 类型的结构体 Ethernet_Data 作为与 RTL8019AS 以太网驱动程序的接口,其中指针 Stptr 指向 IPX 协议层数据包(即 IPX_data),Length 为以太网报头的长度 14,Send_Data 指向以太网报头的地址(即&Ethernet_Header)。

3.3 以太网驱动程序设计

以太网驱动程序是控制 RTL8019AS,实现收发数据。它包括 3 个部分:RTL8019AS 芯片的初始化,接收数据包和发送数据包。分别通过 3 个函数来实现,函数定义如下:

```
InitNic(void); // 初始化程序
Send_Packet(struct _pkt *TxdData); // 发送数据包程序
Rec_Packet(void); // 接收数据包程序
```

初始化是在 RTL8019AS 芯片停止工作的条件下进行的,主要是对 RTL8019AS 芯片内部寄存器进行配置,其中包括了 RTL8019AS MAC 地址的写入和 SRAM 的分频,然后使该芯片进入运行状态。

发送程序编写, Send_Packet()函数中的参数 struct Packet *Txd Data 用于传递需要发送的数据包及数据包的总长度。在 Send_Packet()函数中首先通过参数 TxdData 计算出需要发送的数据的总长度,然后将要发送的数据写入到 RTL8019AS 芯片的发送缓存区,设置芯片的发送起始地址,然后启动发送。如果发送数据失败时可以重新发送,最多重发次数由 MaxSen Number 设定。发送程序的流程如图 3 所示。

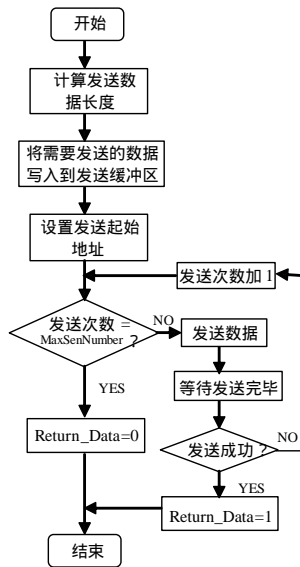


图 3 发送程序流程

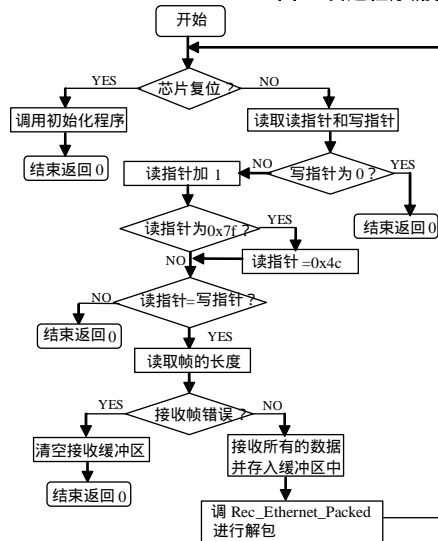


图 4 接收程序流程

接收程序编写,在数据接收程序中,如果接收到的数据存储在接收缓冲区中没有被 ARM7 及时地读取,新接收到的数据会将还没来得急处理的数据覆盖掉,这样将导致数据包的丢失。解决问题的办法是设置多个接收缓冲区,数据接收的具体流程如图 4 所示。

4 实验结果

在本文设计的自动测量系统中(其 IPX 通信结构如图 1), DSPEC 从外部设备在线采集并存储了 32KB 的实验数据,ARM 得到该批数据的可采集信号后,ARM 通过 IPX 通信协议向 DSPEC 发出数据采集命令和连续 32 个数据续传命令,从而,ARM 通过 IPX 通信从 DSPEC 得到 32KB 的实验数据。经过该系统多次的在线测量实验表明,ARM 与 DSPEC 之间的 IPX 通信过程是高速稳定可靠的,满足了本自动测量系统的需要。

5 结束语

本文结合实际情况,详细地分析了 IPX 通信协议,包括 IPX 协议的数据帧格式及其含义,基于 32 位 ARM 微控制器,编写了 RTL8019AS 的驱动程序和 IPX 数据包的打包与解包程序,成功设计了 32 位 ARM 微控制器的 IPX 通信模块,解决了 ARM 的 IPX 通信问题,为 32 位 ARM 微控制器与具有 IPX 通信接口的测量设备之间的 IPX 通信提供了解决方案,具有一定实用价值。

参考文献

- 1 周立功. ARM 的原理及应用[M]. 北京: 北京航空航天大学出版社, 2003.
- 2 周立功. ZLG/IP 的原理及应用[Z]. 广州周立功单片机发展有限公司, 2003.
- 3 刘化君. IPX 协议分析及应用[J]. 临沂师专学报, 1998, 20(3): 55.
- 4 牟连佳. 以太网和 web 网相结合用于工业控制[J]. 微计算机信息, 2003, 19(1): 26-27.
- 5 夏洪山. Novell NetWare 网络应用软件开发指南[M]. 北京: 科学出版社, 1997.
- 6 美国 ORTEC 公司. DSPEC 操作手册[Z]. 2001.

(上接第 251 页)

6 关于 SMI-S 的应用设想

基于 CIM 和 WBEM 的 SMI-S 模型使系统管理者和管理人员能够采用统一的方式控制来自不同厂家的设备和资源。从这一点上讲,它的主要目标和网络集成、共享资源的目标是一致的。因此,可以分析 SMI-S、CIM 和 WBEM 等标准规范在数据网格中的应用方法,研究利用网络思想和存储规范进行广域存储网络的管理,尝试融合广域 IP 存储网络管理和数据网格实现技术,研究广域数据网格中利用 SMI-S 技术进行拓扑结构发现以及数据、元数据的快速传输和复制管理等技术。

参考文献

- 1 Clark T. 存储区域网络设计——实现光纤通道和 IP SAN 的实用指南[M]. 2 版. 邓劲生, 李宝峰, 李 蕾, 译. 北京: 电子工业出版社, 2005-01.

- 2 SNIA Storage Management Initiative Specification (Version 1.0.2 Final)[Z]. 2004-02. Storage Networking Industry Association (SNIA), <http://www.snia.org>.
- 3 Gelada. HP 收购 AppIQ SRM 市场生变数[Z]. 2005-09-21. <http://www.stor-age.com/dongtai/hangye/htm2005/05092118EEE6.asp>.
- 4 Scheefer M. Innovatives Speichermanagement: IBM Speicher-virtualisierung[Z]. http://www-5.ibm.com/de/mittelstand/region/suedwest/bodensee/pdf_bilder/bodensee_speichermgm.pdf.
- 5 Katzenberger W. EMC ControlCenter and SMI-S[Z]. http://www.emc-special.de/isd04/downloads/EMC_ControlCenter_SMI-S.pdf.
- 6 Wubbolt H. Advanced SAN Architecture & Design, EMEA Network Storage Solutions[Z]. http://accessproducts.nl/sectienieuws/evenementen/StorageSeminar/Presentaties/HP/1_HenkWubbolt_030424_StorageSeminar.pdf.