文章编号:1001-9944(2010)07-0017-04

基于MSP430和DM9000的以太网接口设计

刘亚萍,甄国涌,刘东海
(中北大学 电子测试技术国家重点实验室,太原 030051)

摘要:为提高以太网的数据传输效率,设计了以MSP430FG4618为核心的控制单元,驱动以太网接口芯片DM9000EP的嵌入式系统的以太网接口。通过单片机完成网卡芯片的初始化、数据的封装、接收和发送控制等,而网卡芯片则负责以太网数据的接收和发送。测试结果表明了UDP数据传输过程,经自行裁剪的UDP/IP协议栈,提高了UDP传输效率。该接口具有硬件接口简单、外围器件少、价格低廉、开发周期短等特点,同时也能满足测试、采集等高速数据传输系统的要求。

关键词:MSP430;DM9000;以太网接口;UDP;数据传输
中图分类号:TP393.02　　**文献标志码**:A

Design of Ethernet Interface Based on MSP430 and DM9000

LIU Ya-ping,ZHEN Guo-yong,LIU Dong-hai
(National Key Laboratory for Electronic Measurement Technology,North University of China,Taiyuan 030051,China)

Abstract:In order to improve the Ethernet data transmission efficiency, it designed a embedded Ethernet interface, whose core control unit is MSP430FG4618 to drive the Ethernet interface chip DM9000EP. Through the MCU to completed the initializatioin of the Ethernet card chip, data package, the control of receiving and sending; while the Ethernet card chip is used to receive and send data from Ethernet. The test results showed that the transmission process of UDP data; and UDP transmission efficiency is increased through cutting the UDP/IP protocol stack. With simple design,low price,little peripheral equipment and short development cycle, the interface can satisfy some requires from the testing, data acquisition systems of high speed data transmission.

Key words:MSP430;DM9000;Ethernet interface;UDP;data transmission

随着嵌入式技术和网络技术的迅速发展,以太网接口在嵌入式系统中的应用越来越广泛[1]。与传统的RS-232、RS-485、CAN等相比较,以太网通信更加高速、通用、而且可以直接与Internet相连接,提供更大范围的远程访问[2]。在工业以太网技术发展的今天,可以应用在对通信确定性、实时性、稳定性与可靠性要求都较高的测试系统、数据采集系统、远程控制系统等。本设计使用单片机MSP430FG4618、以太网控制器DM9000和经过自行裁剪的UDP/IP协议栈,构成了嵌入式系统的以太网接口,实现UDP通信。

1 系统方案

单片机控制整个系统的运行,以太网控制器实现网络传输的底层功能。即由MSP430完成网卡驱动——初始化DM9000,将单片机自行组织的数据

收稿日期:2010-03-15;**修订日期**:2010-04-07
基金项目:国家自然科学基金项目(60871041)
作者简介:刘亚萍(1985-),女,在读硕士研究生,研究方向为高速数字量设计;甄国涌(1971-),男,工学博士,副教授,研究方向为嵌入式系统、动态测试等;刘东海(1982-),男,硕士研究生,研究方向为动态测试、高速数据采集。

进行以太网帧封装,发送;对接收到的数据帧,去掉以太网头并进行 CRC 校验，存入接收缓存等。DM9000 接收通过 RJ-45 在远程主机中传来的数据,或发送由单片机接自行组织的数据,再由 RJ-45 传送到远程主机，从而实现对数据的发送与接收。系统框图如图 1 所示。

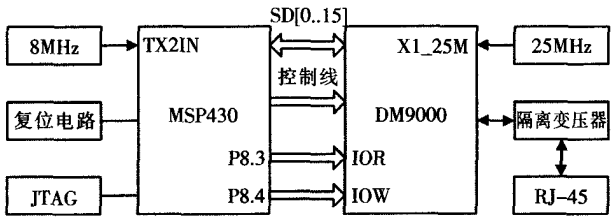

图 1 系统框图

Fig.1 Block diagram of system

系统的网络接口部分主要完成物理层和链路层的功能。单片机作为系统的主控芯片,DM9000 与单片机的连接选择 ISA16bit 模式。数据总线与单片机的 P9~P10 即 PB 相连,CMD 与 P8.1 相连,高为数据控制,低为地址控制。P8.2~P8.7 分别与 INT,IOR,IOW,AEN,WAIT,RST 相连,POW_RST 悬空，用以初始化 DM9000。图 2 为网络接口连接图。

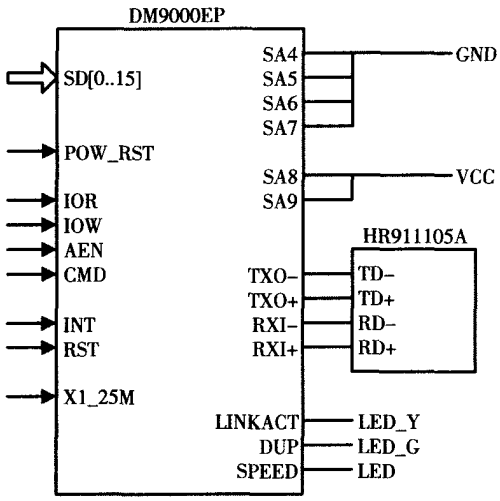

图 2 网络接口框图

Fig.2 Block diagram of Ethernet interface

DM9000 只对接收到的数据进行编码与传送，完成以太网传输。当系统需要和以太网进行通信时，首先通过总线将需要发送的数据写入 DM9000 芯片的内部寄存器 MWCMD 中,然后由 DM9000 自动将数据发送给以太网；当 DM9000 接收到来自以太网的数据时，系统通过总线方式将数据读入到 MRCMD 中。DM9000 工作的默认基地址为 0x300,这里按照默认地址选择,将 SA8、SA9 接高电平(3.3V),SA4~SA7 接地。0x800300 是 DM9000 命令端口的地址,对它的赋值操作就相当于把数据写到该地址中,即把数据写到 DM9000 的命令端口中。

HR91105A 是内嵌隔离变压器的 RJ-45。隔离变压器是以太网接口实现中必不可少的。它的主要功能是转换电平并抑制高频干扰接入以太网,防止烧坏元器件,实现带电插拔功能。RJ-45 接口实现终端设备接入 Internet 或远程 PC 机。该器件的选择简化了器件之间的连线,节省了 PCB 板的空间,同时也提高了高频信号传输的可靠性。

2 DM9000 驱动程序

2.1 DM9000 初始化

初始化时要进行以下的操作来使 DM9000 芯片达到工作状态:

(1)开启 DM9000 芯片,DM9000 芯片上电默认的状态是 GEPIO0=1 的 Powerdown 状态，所以要将 GEPIO0 bit[0]设置为 0,来打开 DM9000 芯片。上电之后将 GPCR(REG_1E)GEP_CNTL0 bit[0]设置为 1;同时将 GPR(REG_1F)GEPIO0 bit[0]设置为 0 来打开芯片;

(2)进行两次软启动,根据芯片的设计要求,要使芯片达到工作状态,在上电之后就要对芯片进行两次软启动，软启动是通过设置 NCR(REG_00)bit[2:0]=011 (至少延时 20μs)，设置 NCR(REG_00) bit[2:0]=000 来实现的,同样的操作要进行两次;

(3)清除 Tx Status 寄存器;

(4)设置 IMR(REG_FF)寄存器 PRM bit[0]/PTM bit[1]开启 TX/RX 中断;

(5)设置 RCR 寄存器来使能 RX。RX 功能函数的使能是靠设置 RX 控制寄存器(REG_05)RXEN bit[0]=1。

当进行了以上的步骤之后,DM9000 芯片就处于工作状态。当由于异常而引发重启时就要再次进行相同的操作过程来使 DM9000 芯片恢复到正常状态。

2.2 发送函数

在传送 1 个封包之前,需将其封包资料存放在 DM9000 的传送内存中 0000h-0BFFh。若是写入位置超过 0BFFh 时,DM9000 会自动将位置移到 0000h 的位置。将封包资料存放在 MWCMD 中，

DM9000 会自动将其资料存向其传送内存中。另外还需将要传送封包的大小存放在 TXPLL 和 TXPLH。之后再将 TCR 的 bit0 设为 1，此时开始进行封包的传送。而在传送完成后，会将传送是否成功的信息放在 TSRI，TSRII 中。放的顺序为 TSRI->TSRII->TSRI->TSRII，所以需要依照 NSR 的 bit2-3 来判断现在是 TSRI 或 TSRII 传送完成[3-4]。发送流程图如图 3 所示。

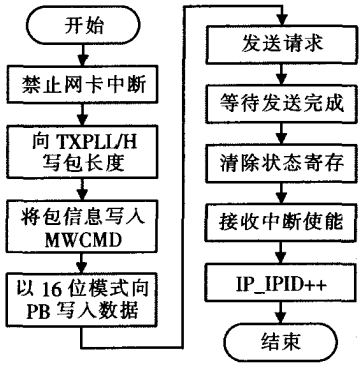

图 3 发送流程图

Fig.3 Sending's flow chart

发送函数：Send_Package (INT8U *senddata, INT16U datalength)

```
DM9000_Reg_Write(IMR,0x80);//禁止网卡中断,防止在发送数据时被中断干扰
DM9000_Reg_Write (TXPLH, ((len >>8)&0x0ff));//发送包长度高 8 位
DM9000_Reg_Write(TXPLL,(len&0x0ff));
//发送包长度低 8 位
PBOUT=MWCMD;
.....
for(i=0;i<len;i+=2)//16 bit mode
{
    Delay_N_mS(4);
    PBOUT=senddata[i](INT16U)(senddata[i+1]<<8);
    .....
}
DM9000_Reg_Write (TCR,0x01);//发送请求,发送完毕 bit0 自动清零
while ((DM9000_Reg_Read (NSR)&0x0c) == 0);//等待数据发送完成
Delay_us(2);
DM9000_Reg_Write(NSR,0x2c);//清除状态寄存器
DM9000_Reg_Write (IMR,0x81);//DM9000 接收中断使能
IP_IPID++;
```

2.3 接收函数

数据包接收功能是 DM9000 芯片实现网络功能的基础，接收流程如图 4 所示。在接收数据时采用中断方式，即当有数据到来并在 DM9000 内部 CRC 校验通过后会产生一个接收中断，中断发生时可以将 DM9000 所接收到的数据包读出并交由上层协议进行处理。接收到的数据在经过了硬件部分的 CRC 校验之后存放在 RX FIFO 中，在 DM9000 中的内部地址 0x0C00-0x3FFF(13K byte)。DM9000 接收到一个数据包后，会在数据包前面加上 4 个字节，分别为“01H”、“status”(同 RSR 寄存器的值)、LENL (数据包长度低 8 位)、LENH(数据包长度高 8 位)。其中“01H”表示接下来的是有效包数据。这一位为接收数据标志，如果该位是 01 则表示有数据被接收且保存到 RX SRAM 中，在 DM9000 中的内部地址 0x0C00-0x3FFF(13K byte)；如果该位为 00 则表示没有数据到达，中断可以直接返回；如果既不是 01 又不是 00，则认为有异常发生，这时就要将 DM9000 芯片重启以使芯片恢复到正常状态。第二个字节是

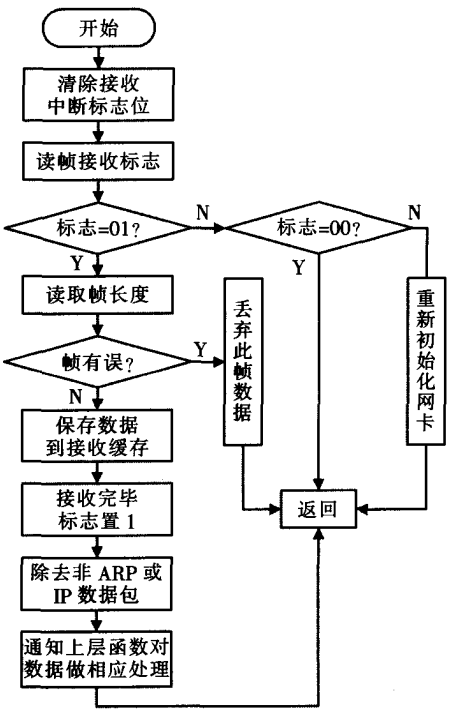

图 4 接收流程图

Fig.4 Receiving's flow chart

status,这个字节是接收数据包的状态字,其中的内容与接收状态寄存器(REG_06H)中的内容相同,可以用来判断所接收的数据包是否正常,或发生了何种异常,这样就可以针对不同的异常进行不同的操作,实现对接收任务的控制。第3字节、第4字节存有接收到的数据包的长度,在读取数据包的时候要用这个长度来进行控制。这四个字节是DM9000在接收数据的时候添加的信息,不属于数据包的内容。从第5个字节开始的数据才是真正的数据包的内容,长度存在第3、4字节当中,所以驱动程序可以依次读取数据包数据,直至达到数据包长度结束[3,5]。以下为接收函数。

```
接收函数:Receive_Package(INT8U *recvdata)
if(DM9000_Reg_Read(ISR) & 0x01)
{
    DM9000_Reg_Write (ISR, 0x01);//清除接收中断标志位
}
ready= DM9000_Reg_Read(MRCMDX);//第一次读取,一般读取到的是 00H
if((ready & 0x0ff) ! = 0x01)
{
    ready= DM9000_Reg_Read(MRCMDX);//第二次读取,总能获取到数据
    ……
}
status = DM9000_Reg_Read(MRCMD);
if(! (status & 0xbf00) && (len < 1522))
{
    ……
    recvdata[i] = tem & 0x0ff;
    recvdata[i + 1] = (tem >> 8) & 0x0ff;
}
Receive_finish=1;
……
```

3 系统UDP通信测试

用wirshark网络分析软件抓取的UDP通信测试过程,如图5所示。

前三个ARP请求是因为程序在运行时在发送UDP包前遇到的断点,wirshark处于捕获状态,因此捕获ARP包信息,当驱动程序全速运行之后,wirshark抓取到的全部为UDP包。证明测试结果正确。以下对捕获到的每8帧数据(如图6)进行分析。

No.	Time	Source	Destination	Protocol	Info
1	0.000000	Giga-Byt_2c:66:c6	Broadcast	ARP	Gratuitous ARP for 192.168.201.112 (Request)
2	0.014996	Giga-Byt_2c:66:c6	Broadcast	ARP	Gratuitous ARP for 192.168.201.112 (Request)
3	1.015024	Giga-Byt_2c:66:c6	Broadcast	ARP	Gratuitous ARP for 192.168.201.112 (Request)
4	21.442870	192.168.201.5	192.168.201.112	UDP	Source port: boks Destination port: socalia
5	21.443607	192.168.201.5	192.168.201.112	UDP	Source port: boks Destination port: socalia
6	21.444896	192.168.201.5	192.168.201.112	UDP	Source port: boks Destination port: socalia
7	21.446620	192.168.201.5	192.168.201.112	UDP	Source port: boks Destination port: socalia
8	21.447908	192.168.201.5	192.168.201.112	UDP	Source port: boks Destination port: socalia
9	21.449628	192.168.201.5	192.168.201.112	UDP	Source port: boks Destination port: socalia
10	21.451349	192.168.201.5	192.168.201.112	UDP	Source port: boks Destination port: socalia
11	21.452638	192.168.201.5	192.168.201.112	UDP	Source port: boks Destination port: socalia
12	21.454360	192.168.201.5	192.168.201.112	UDP	Source port: boks Destination port: socalia

图5 UDP通信测试过程

Fig.5 Process of the UDP communication test

```
0000  00 24 1d 2c 66 c6 09 08  07 05 04 03 08 00 45 00   .$.,f... ......E.
0010  00 96 00 0a 00 00 80 11  26 86 c0 a8 c9 05 c0 a8   ........ &.......
0020  c9 70 19 64 13 ec 00 82  a5 75 30 31 32 33 34 35   .p.d.... .u012345
0030  36 37 38 39 3a 3b 3c 3d  3e 3f 40 41 42 43 44 45   6789:;<= >?@ABCDE
0040  46 47 48 49 4a 4b 4c 4d  4e 4f 50 51 52 53 54 55   FGHIJKLM NOPQRSTU
0050  56 57 58 59 5a 5b 5c 5d  5e 5f 60 61 62 63 64 65   VWXYZ[\] ^_`abcde
0060  66 67 68 69 6a 6b 6c 6d  6e 6f 70 71 72 73 74 75   fghijklm nopqrstu
0070  76 77 78 79 7a 7b 7c 7d  7e 7f 80 81 82 83 84 85   vwxyz{|} ~.......
0080  86 87 88 89 8a 8b 8c 8d  8e 8f 90 91 92 93 94 95   ........ ........
0090  96 97 98 99 9a 9b 9c 9d  9e 9f a0 a1 a2 a3 a4 a5   ........ ........
00a0  a6 0a eb 90                                       ....
```

图6 捕获到的UDP数据包数据显示

Fig.6 Show of the eithth captured UDP

以上数据中,"00 24 1d 2c 66 c6"为目的地址,"09 08 07 05 04 03"为源地址,"80 00"为帧类型(IP数据包,大端模式),"45 00"为协议类型(UDP协议),"30 31……eb 90"为发送的测试数据,符合以太网数据帧格式。

4 结语

该接口采用TI公司的MSP430微处理器和以太网协议芯片相结合,组成了嵌入式以太网接口,通过软件实现了简单的UDP/IP协议,成功地将曾经相对独立的测试系统与远程PC机连结起来,在8MHz主频的MCU得到较高的利用,有效地提高了UDP数据传输速率。同时为嵌入式测试系统通过网络进行远程登陆、访问、采集、监控等提供了可能,这也正符合当今嵌入式设备趋于网络化发展的方向。

参考文献:

[1] 周晓峰,杨世锡,华亮.单片机上简TCP/IP协议的实现[J].微电子学与计算机,2004,21(2):99-101.

[2] 王飞石,广田.基于ARM和DM9000的网卡接口设计与实现[J].ARM开发与应用,2008,24(5):123-125.

[3] 姜小涛.基于ARM和以太网的数据采集系统设计与实现[D].大连海事大学,2003,6.

[4] 秦龙.MSP430单片机C语言应用程序设计实例精讲[M].北京:电子工业出版社,2006.

[5] 兰少华,杨余旺,吕建勇.TCP/IP网络与协议[M].北京:清华大学出版社,2006.