

编程修改网卡 MAC 地址的方法

上海交通大学机电控制研究所 吕刚 周爱国

摘要：本文针对 RELTECK 公司的 RTL8029AS 芯片介绍了一种直接修改网卡 MAC 地址的方法，程序用 C 语言与汇编语言混合编程实现，并在实践中检验通过。

关键词：网卡 MAC C 语言 混合编程

每一块网卡都有一个全球唯一的 MAC 地址，在网络通信中起到区分每一个节点的作用，因为它的唯一性，也常常被用作每台计算机的身份标志。对于目前市场上广为使用的基于台湾 RELTECK 公司芯片的 PCI 网卡来说，网卡的 MAC 地址存储在网卡上的芯片 93C46 中，这是一块可读写的 128 字节 EEPROM，每当网卡重启时，网卡上的主控芯片 RTL8029AS 就会从 93C46 中读取网卡的 MAC 地址，放到主控芯片中 RAM 中，但是这并不意味着网卡就会以该 MAC 地址工作，网卡工作时所用的 MAC 地址是可以由系统设定的，一般情况下，系统总是简单地将网卡工作时的 MAC 地址设为 93C46 中的 MAC 地址，相信不少人已经掌握了在系统中设置网卡工作 MAC 地址的方法，但这不是这里要讲的内容。我们可以用软件方法从根本上改变 93C46 中存储的 MAC 地址，事先需要了解 PCI 网卡和 BIOS 调用的一些技术内容。

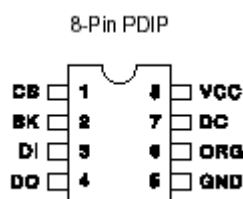


图 1 93C46 芯片引脚分配

一. PCI 网卡的配置空间与 I/O 空间

1. 网卡配置空间

配置空间是 PCI 总线设备实现 PNP 功能的基础，它是一容量为 256 字节(从 00H 到 FFH)并具有特定记录格式的地址空间，在没有对 PCI 设备的配置空间进行配置的情况下，是不能对 PCI 设备的 I/O 空间和存储器空间进行读写的。在 PCI 网卡配置空间中，有三个非常重要的字段，一个是 Vendor ID (01-00H; 只读)，它是 RTL8029AS 主控芯片的制造商的标志，每个制造商都有全球唯一的标志，REALTEK 公司的标志是 10ECH; 另一个是 Device ID (03-02H; 只读)，它是 REALTEK 底下产品的标志，可想而知 RTL8029AS 主控芯片的标志是 8029H; 还有一个是 I/O 空间的基址 (13-10H, 可读写)，它是一个双字，最后四位不能写，读出总是 1H，表示是 I/O 地址空间的基址。当计算机启动时，BIOS 会自动对所有 PCI 设备的配置空间进行配置，合理分配 I/O 空间和存储器空间，使它们在空间地址上不重叠。

Vendor ID 与 Device ID 是我们用来搜索 PCI 设备的关键字段,通过它我们可以知道特定的 PCI 设备是否存在,以及该设备的总线号,设备号与功能号,有了这些信息,我们就可以调用 BIOS 中断函数对 PCI 设备的 I/O 空间进行读写了。

2. 网卡 I/O 空间

PCI 网卡的 I/O 空间长度为 1FH,如果网卡的 I/O 空间基址为 E800H,则网卡 I/O 空间的地址就为 E800H 到 E81FH。实际上,从 00H 到 1FH 有四页 (page0-page3) 在地址上重合内容,页之间的切换是通过 00H 处的寄存器 CR 来实现的。所有的四页寄存器都可以看作是 RTL8029AS 芯片的工作寄存器,它决定了网卡的工作方式,也用于网络数据的输入与输出。其中前三页寄存器是每种网卡都必须实现的,它是 NE2000 标准的内容,而的四页的内容因芯片厂家而异。我们所关心的与 93C46 相关的寄存器 93C46CR 就是位于第四页上,偏移为 01H,在程序中用 N93C46CR 表示。

下面表一与表二分别是 CR 寄存器与 93C46CR 寄存器的说明。

表一 CR 寄存器功能描述

位	符号	功能描述
7-6	PS1, PS0	00-11H分别表示page0-page1
5-2	RD2-0, TXP	不关心
1-0	STA, STP	STA=1, STP=0时工作; STA=0, STP=1时停止

表二 9346CR 寄存器功能描述

位	符号	功能描述
7-6	EEM1-0	10B表示9246编程模式
5-4	-	不用
3	EECS	对应9346 CS脚
2	EESK	对应9346 SK脚
1	EEDI	对应9346 DI脚
0	EEDO	对应9346 DO脚

由表二可以知道,9346CR 寄存器的低四位是与 93C46 芯片上的管脚一一对应的,所以我们可以通过读写 9346CR 寄存器来间接地读写 93C46 存储器,从而达到修改 MAC 地址的目的。具体读写方式需要遵从 93C46 的读写规范。需要指出的是,网卡中 93C46 的存储空间是按字分配的,即分配为 64 个字,且低字节在前。

二. 程序实现

1. BIOS 功能调用

BIOS 功能调用主要用于对网卡配置空间的读写上,由配置空间的数据我们可以得到

RTL8029AS 芯片的网卡是否存在的信息，最关键的是可以得到网卡 I/O 空间的基址。我们将要用到 BIOS 中断 1AH 的两种功能服务，分别如下表所示：

查找PCI设备		
入口 参数	[AH]	PCI_FUNCTION_ID
	[AL]	FIND_PCI_DEVICE
	[CX]	Device ID
	[DX]	Vendor ID
	[SI]	索引号
出口 参数	[BH]	总线号
	[BL]	设备功能号
	[AH]	返回状态
	[CF]	出错状态

表三 查找 PCI 设备 BIOS 功能调用

读配置空间		
入口 参数	[AH]	PCI_FUNCTION_ID
	[AL]	READ_CONFIG_WORD
	[BH]	总线号
	[BL]	设备功能号
	[DI]	寄存器号
出口 参数	[CX]	读出的字
	[AH]	返回状态
	[CF]	出错状态

表四 读 PCI 设备配置空间功能调用

其中 PCI_FUNCTION_ID、FIND_PCI_DEVICE、READ_CONFIG_WORD 都是系统常数，Device_ID 和 Vendor_ID 在这里分别为 8029H 与 10ECH。

2. 部分子程序

下面给出确定 PCI 网卡 I/O 地址空间基址的汇编子程序部分代码，程序采用嵌入式汇编方式，其它还有些判别与保护代码此处略去，系统参数可参考有关文献，该程序段将读出的 I/O 基址的低字保存在变量 readword0 中，高字保存在变量 readword1 中。

```
asm mov  ah,PCI_FUNCTION_ID      /*查找 PCI 网卡*/
asm mov  al,FIND_PCI_DEVICE
asm mov  cx,8029h                /*Device ID*/
asm mov  dx,10ech                /*Vendor ID*/
asm mov  si,0                    /*索引号为 0*/
asm int  1Ah
asm mov  temp,bl                 /*保存设备功能号*/
asm mov  ah,PCI_FUNCTION_ID      /*读 I/O 基址低字*/
```

```

asm mov  al,READ_CONFIG_WORD
asm mov  di,10h
asm int  1Ah
asm mov  readword0,cx
asm mov  ah,PCI_FUNCTION_ID      /*读 I/O 基址高字*/
asm mov  di,12h
asm int  1Ah
asm mov  readword1,cx

```

以下是对 93C46 写的子程序，其中 CR 的地址就是上面得到的基址，而 N9346CR 的地址为 CR+1，必须注意的是在每个 outportb 和 inport 子函数之间需要加上一定的延时，因为计算机的速度远比 93C46 的读写速度要快，延时是为了保证速度匹配，程序中为了节省篇幅没有列出，一般 5ms 左右就可保证读写正常。

```

void do_93c46(uchar address)
{
    uchar count;
    for(count=0;count<8;count++)
    {
        if((address & 0x80)!=0)
        {
            outportb(N9346CR,0x8a);
            outportb(N9346CR,0x8e);
            outportb(N9346CR,0x8a);
        }
        else
        {
            outportb(N9346CR,0x88);
            outportb(N9346CR,0x8c);
            outportb(N9346CR,0x88);
        }
        address=address<<1;
    }
}

void write_93c46(uchar address,uint value)
{
    uint temp;
    temp=value & 0xff00;
    temp=temp>>8;
    value=value<<8;
    value=value | temp;
    outportb(CR,0xe2);
    outportb(N9346CR,0x8a);
    outportb(N9346CR,0x8e);
    do_93c46(0x3c);
}

```

```

outportb(N9346CR,0x88);
outportb(N9346CR,0);
outportb(N9346CR,0x8a);
outportb(N9346CR,0x8e);
address=address | 0x40;
do_93c46(address);
do_93c46(value>>8);
do_93c46(value&0xff);
outportb(N9346CR,0x88);
outportb(N9346CR,0);
outportb(N9346CR,0x88);
for(value=0;value<1000;value++)
{
    savedata=inport(N9346CR);
    if(savedata & 0x01)break;
}
outportb(N9346CR,0);
outportb(N9346CR,0x8a);
outportb(N9346CR,0x8e);
do_93c46(0);
outportb(N9346CR,0x88);
outportb(N9346CR,0);
}

```

参考文献:

- [1] RTL8029AS Realtek PCI Full-Duplex Ethernet Controller with built-in SRAM Advance Information.
- [2] PCI BIOS Specification Revision 2.1
- [3] AT93C46 Specification. www.Atmel.com
- [4] www.laogu.com