

# 嵌入式系统的以太网接口开发

王 馨,高文华,董增寿

(太原科技大学 电子信息工程学院,山西 太原 030024)

**摘 要:**随着网络技术的迅速发展,嵌入式系统也向着网络化的方向发展。而以太网的广泛使用,使得嵌入式技术与以太网结合就成为一个新的发展研究方向。系统采用三星 S3C2440 嵌入式微处理器,CS8900A 以太网控制芯片,分析了以太网芯片软件设计的主要步骤,重点分析其原理及实现过程,最后将驱动程序采用静态编译的方式编译进内核,至此完成了在 Linux 操作系统下以太网的驱动开发。实现了嵌入式设备更高可靠性以及更快速度的通讯方式,提高了嵌入式系统的整体性能。

**关键词:**嵌入式技术;以太网;S3C2440;CS8900A;Linux 操作系统;驱动

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2012)06-0210-04

## Ethernet Interface Development of Embedded System

WANG Xin, GAO Wen-hua, DONG Zeng-shou

(Electronic Information Engineering College, Taiyuan University of Science and Technology, Taiyuan 030024, China)

**Abstract:** Along with the rapid development of network technology, embedded system is developing toward the direction of network. While Ethernet is widely used, embedded technology and Ethernet combination became a new development direction of research. The system adopts the Samsung S3C2440 embedded microprocessor, Ethernet control chip using CS8900A, analyses the main steps of the Ethernet chip software design, focuses on the analysis of its principle and realization process. Finally the driver compiled into the kernel with a static way, this completed the Ethernet driver development in Linux operating system. Realize higher reliability and faster means of communication in embedded equipment, improve the overall performance of the embedded system.

**Key words:** embedded technology; Ethernet; S3C2440; CS8900A; Linux operating system; driver

## 0 引言

嵌入式系统是一种资源受限的专用计算机系统,灵活的应用是其很大的一个优势,是一个新的技术发展方向。目前已经广泛地应用在诸如消费电子、网络通信、工业控制<sup>[1]</sup>等各种领域,因而具有很好的市场前景。随着网络技术的发展,嵌入式系统的网络接口开发就显得尤为重要。因此嵌入式技术与网络技术的结合有很重要的实际意义。

## 1 系统硬件选择

嵌入式系统通常是面向特定应用的,所以针对不同的具体环境而设计的嵌入式系统之间的差别也很大<sup>[2]</sup>。需要考虑处理器的选择、网络接口及用户接口。

系统框图如图1所示。

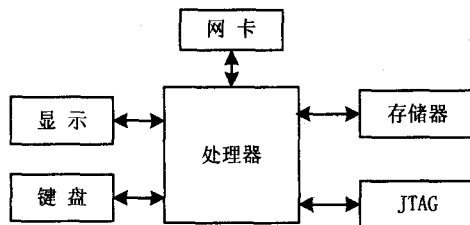


图1 系统框图

### 1.1 系统嵌入式微处理器的选择

嵌入式系统设计的关键是选择一款合适的嵌入式微处理器,在选择过程中,需要考虑技术的先进性、音频与视频的可扩展性以及应用中的可靠性等诸多方面的因素。而考虑到所需要的功能、处理的速度、寻址的能力、片上所集成的接口、性价比等各个方面,本系统选择三星公司的一款基于 RM920T 内核的芯片——S3C2440<sup>[3]</sup>。图2为S3C2440片上结构原理图。

三星 S3C2440 处理器有着相对丰富的集成接口和其他一些资源<sup>[4]</sup>,例如:外部存储控制器,支持 Nand flash 系统启动;2个USB主机接口,1个USB设备接口;片内锁相环,主频 400MHz,可倍频至 533MHz;117

收稿日期:2011-11-15;修回日期:2012-02-21

基金项目:国家自然科学基金(4114026)

作者简介:王 馨(1982-),女,山西神池人,硕士研究生,研究方向为智能信息处理;高文华,教授,硕士生导师,研究方向为智能信息处理;董增寿,教授,硕士生导师,主要研究方向为信号检测与模式识别、远程故障诊断。

个多功能 I/O 接口;16K 指令缓存/16K 数据缓存/MMU;Camera 接口和实时时钟。选择 ARM 处理器的另一个比较客观的原因是,现在 ARM 处理器的市场占有率逐渐增高,可以很方便地实现系统的扩展。

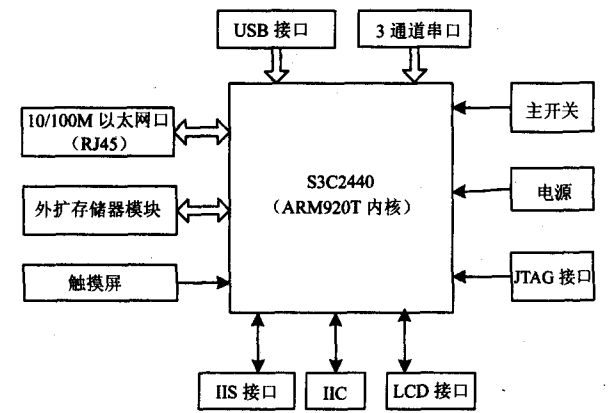


图 2 S3C2440 片上结构原理图

1.2 系统以太网控制芯片的选择

随着以太网技术的快速发展和在不同行业的大量使用,使得各种新的以太网芯片不断出现,而 CS8900A 就是其中性价比比较好的一种。通过实践应用可以说明此芯片性能优异,可以稳定实现以太网的连接<sup>[5]</sup>。

CS8900A 芯片是一个 16 位以太网控制芯片,能耗低且性能稳定是其重要的特点,在嵌入式领域中使用非常常见。此芯片的重要的特点就是可以根据实际需要进行动态调整数据的传输和工作模式,同时还能设置内部寄存器值以适应不同的工作环境。

此芯片符合 IEEE 802.3 的以太网标准;有 4K 的片内 RAM;具有 I/O 模式、存储器模式和 DMA 操作模式三种工作模式;自动生成报头、自动进行 CRC 检验、冲突后自动重发;支持全双工操作;支持外部的 EEPROM。其功能框图如图 3 所示。

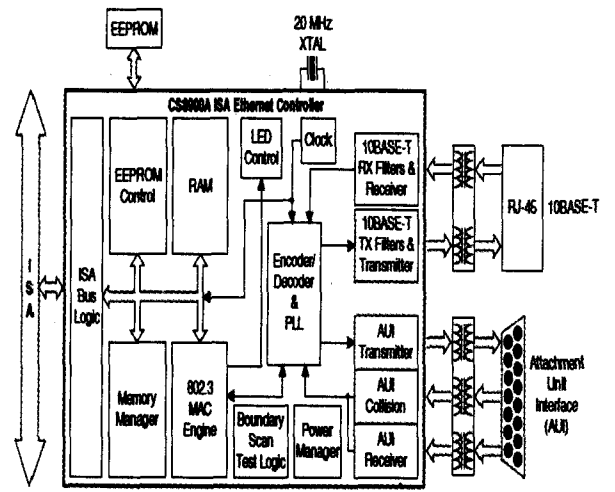


图 3 CS8900A 以太网芯片功能框图

其基本工作原理为:当设备收到主机发过来的数据报后,监听网络线路。如果线路空闲,则立即发送该

数据帧。如果遇线路忙,就一直等到空闲为止发送。在发送过程中,首先要添加以太网帧头,其次待 CRC 校验码生成后将此数据帧发送到以太网上。再次是数据的接收,将需要经过解码、去掉以太网帧头和地址检验等一系列处理后的数据帧缓存到片内存储。然后是 CRC 校验,并告知主机设备收到了数据帧,最后传输到主机的存储区内。

1.3 硬件连接及软件设计

文中外扩一个 CS8900A 以太网控制芯片<sup>[6,7]</sup>,用于补充 S3C2440 处理器所没有的网络功能。

CS8900A 与 S3C2440 的电路连接图如图 4 所示。

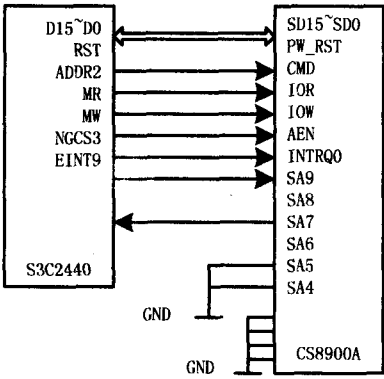


图 4 CS8900A 与 S3C2440 的电路连接图

2 以太网芯片软件设计

CS8900A 芯片的软件设计<sup>[6-12]</sup>主要有 6 个部分:

(1)CS8900A 初始化。

设备初始化是一个关键步骤,也是以下所有环节的基础。文献[9]中对初始化过程做了具体描述,在此不再赘述。如果存在就对设备所需要的资源进行配置,最后在内核中注册该函数并申请所需的内存地址空间。

```
struct net_device
{
    char name[ IFNAMSIZ];    //设备名字
    unsigned long base_addr;  //I/O 地址
    unsigned int irq;        //中断号
    struct net_device * next;
    //指向下一个网络设备,用于维护链表
    int ifindex;             //设备标识符
    struct net_device_stats * (* get_stats)
    (struct net_device * dev);
    //返回一个结构体,统计接收到或发送的信息
    unsigned long trans_start;
    //记录最后一次发送成功的时间
    unsigned short type;     //物理硬件类型
    void (* (* uninit) )(struct net_device * dev);
```

```

//释放网络设备时执行
int (*open)(struct net_device * dev); //打开网络设备
int (*stop)(struct net_device * dev); //停止网络设备
int (*hard_start_xmit)(struct sk_buff * skb, struct net_device * dev); //硬件开始传输
void (*tx_timeout)(struct net_device * dev);
//传输超时
int watchdog_timeo; //看门狗
}

```

#### (2) 设备打开函数。

在激活网络设备时调用 open 函数打开设备。首先在内核中注册中断处理函数; 然后对设备的配置寄存器和控制寄存器进行设置; 开启设备接口传送队列。

```
set_irq_type(dev->irq, IRQT_RISING); //设置中断类型
```

```
cs8900_set(dev, PP_RXCTL, RxOKA | IndividualA | BroadcastA);
```

```
//置位接收控制寄存器
```

```
cs8900_set(dev, PP_TxCFG, TxOKiE | Out of windowiE | JabberiE);
```

```
//置位传送配置寄存器
```

```
cs8900_set(dev, PP_BusCTL, EnableRQ);
```

```
//置位总线控制寄存器
```

```
if((res = request_irq(dev->irq, &cs8900_interrupt, 0, dev->name, dev)) < 0)
```

```
//向内核注册网络设备的中断处理函数
```

```
netif_start_queue(dev);
```

```
//开启设备接口传送队列
```

#### (3) 设备关闭函数。

设备关闭时调用 stop 函数用来释放所占有的系统空间。将所有配置寄存器和控制寄存器的值都设为零, 释放设备中断资源; 终止设备接口传送队列。

```
cs8900_write(dev, PP_BusCTL, 0);
```

```
cs8900_write(dev, PP_TestCTL, 0);
```

```
cs8900_write(dev, PP_SelfCTL, 0);
```

```
cs8900_write(dev, PP_LineCTL, 0);
```

```
cs8900_write(dev, PP_BufCFG, 0);
```

```
cs8900_write(dev, PP_TxCFG, 0);
```

```
cs8900_write(dev, PP_RxCTL, 0);
```

```
cs8900_write(dev, PP_RxCFG, 0);
```

```
free_irq(dev->irq, dev); //关中断
```

```
netif_stop_queue(dev);
```

```
//停止接口传送队列
```

#### (4) 数据发送函数。

在加载设备驱动时, 调用设备的初始化函数进行初始化。成功操作后, 调用打开函数将设备打开, 再建立硬件包头信息, 然后将数据发送出去。

数据的发送过程可分为以下几个步骤:

①在寄存器 TxCMD 中写入传送数据命令且在寄存器 TxLength 中写入待发送数据帧长度。

②若总线状态寄存器 BusST 的 TxBidErr 寄存器置位, 那么数据帧长度非法。若总线状态寄存器 BusST 的 Rdy4TxNOW 置零位, 则此时设备无法接收。记录错误信息。

③将数据送入设备并发送到以太网上。此时释放缓存。

```
cs8900_write(dev, PP_TxCMD, TxStart(AlterS));
```

```
//写传送数据命令控制字
```

```
cs8900_write(dev, PP_TxLength, skb->len);
```

```
//写待发送数据帧长度
```

```
status = cs8900_read(dev, PP_BusST);
```

```
//读取总线状态寄存器
```

```
cs8900_frame_write(dev, skb);
```

```
//将待发数据写入 CS8900A
```

```
dev_kfree_skb(skb); //释放缓存
```

#### (5) 接收函数。

接收数据是通过网络驱动的中断处理函数调用实现网络数据从设备读出。当数据接收到后便会产生一个中断, 同时在中断处理程序驱动中申请一块缓存区用以存放从硬件读出的数据。最后调用 netif\_rx() 函数并把数据送给协议层。数据的读取过程可分为以下几个步骤:

①读出 RxLength 和 RxStatus 的状态值, 获取接收数据帧的数据长度和状态信息。根据数据帧状态信息判断数据接收是否正常, 如果接收正常就继续; 否则函数返回并记录相关错误信息。

②将数据传送到申请的内存缓存区中。

③调用 netif\_rx() 将接收到的数据送往上层协议进行处理。

```
status = cs8900_read(dev, PP_RxStatus);
```

```
//读取状态信息
```

```
len = cs8900_read(dev, PP_RxLength);
```

```
//读取长度
```

```
skb->dev = dev;
```

```
//将网络接口的指针赋给该成员变量
```

```
cs8900_frame_read(dev, skb, length);
```

```
//将数据从片内存储送到缓冲区
```

netif\_rx(skb); //将缓存区中的数据送往上层协议处理

#### (6) 中断处理函数。

中断处理函数通过读取 ISQ 寄存器的内容,来判断触发中断的事件类型以采用不同的处理手段。中断处理函数主要处理像发送事件、接收事件、缓存事件、接收数据包丢失或计数发送数据包碰撞计数等事件。

### 3 网络功能

文中设备网络驱动程序在 AKAES3C2440 集成开发平台上实现,由于网络功能的频繁使用,文中采用了将驱动程序静态编译进内核的方式,以减少模块插入所带来的麻烦。将编译生成的内核映像文件烧写到开发板上,再次启动开发板。设置目标板的 IP 地址为 192.168.1.100,在 Linux 开发环境下的主机终端的 IP 地址设置为 192.168.1.21。然后在开发板上执行 ping 192.168.1.21,在 Linux 环境下的主机上执行 ping 192.168.1.100,检测网络。如果两边都能 ping 通,则网络功能实现,说明开发板实现了网络传输功能。

### 4 结束语

文中在研究嵌入式系统的基础上实现了其以太网接口的开发,基于 S3C2440 的嵌入式平台,作为 ARM 的主流产品,不论从功能、价格还是应用前景上都好于现在所使用的单片机系统,有更广泛的使用领域。在现有功能的基础上,可以把该系统嵌入到其他系统中,实现更多的功能,具有很好的发展前景。

(上接第 209 页)

```
m_2Draw.DeleteFirstPoint();
for(i=0;i<=98;i++)
buffer[i]=buffer[i+1];
uffer[99]= Receiver;
for(i=1;i<100;i++)
m_2Draw.AppendPoint(i, 25+buffer[i]/1000);
}
```

其中 Receiver 定义为从总线收发器接收的数据,以数组的形式表示。

### 4 结束语

文中设计实现的阵列电机控制系统利用了 CAN 总线实现对无刷直流电机的“步进控制”,其结构精巧、工作可靠。同时,设计并完成了—个较为典型的电机阵列人机交互界面,美观大方、交互性强。可以实际应用于具有多点、相关、交互特征的复杂控制系统。因此具有较大的实用与研究价值。

#### 参考文献:

[1] 敖庆荣,袁 坤.伺服系统[M].北京:航空工业出版社,

#### 参考文献:

- [1] 庄晓燕,周森鑫.工业控制以太网协议实现研究[J].计算机技术与发展,2009,19(12):244-247.
- [2] 姜余祥,周小龙.基于 ARM 的嵌入式网关设计与实现[J].计算机科学,2010,37(9A):38-41.
- [3] 吴国庆.对等网络技术研究[J].计算机技术与发展,2008,18(7):100-103.
- [4] Samsung Electronics. S3C2440A 32-Bit CMOS Microcontroller User's Manual, Revision 1[M]. [s. l.]: Samsung Electronics, 2004.
- [5] 黄国刚,陈 明,许海岗,等.基于 CS8900A 的工业级嵌入式网络接口实现[J].计测技术,2005,25(1):16-18.
- [6] 郭建磊.基于 S3C2440 和 Linux 的嵌入式网络驱动程序开发[D].青岛:青岛大学,2009.
- [7] 俞 露.基于 ARM 的嵌入式系统硬件设计[D].杭州:浙江大学,2003.
- [8] 刘 森.嵌入式系统接口设计与 Linux 驱动程序开发[M].北京:北京航空航天大学出版社,2006.
- [9] 杨 路.基于 CS8900A 以太网驱动程序的设计[J].微计算机信息,2009,25(23):61-63.
- [10] Yaghmour K. Building Embedded Linux Systems[M]. USA: O'Reilly, 2003.
- [11] Schulze B. Embedded test offers unique value for serial I/O[M]. [s. l.]: Agilent Technologies, Inc., 2007:25-29.
- [12] 孙天泽,袁文菊,张海峰.嵌入式设计及 Linux 驱动开发指南—基于 ARM9 处理器[M].北京:电子工业出版社,2005.

2006.

- [2] 欧海平.基于 SOPC 的步进电机的减速 PWM 控制器 IP 核设计[J].机电工程,2011,28(6):708-711.
- [3] 张 深.直流无刷直流电动机原理及应用[M].北京:机械工业出版社,1997:2-13.
- [4] 刘 刚,王志强,房建成.永磁无刷直流电机控制技术与应用[M].北京:机械工业出版社,2008:3-10.
- [5] Samsung Electronics Co. The S3C6410A 32bit RISC Microprocessor User Manual[M]. [s. l.]: [s. n.], 2006.
- [6] 王姜博.CAN 节点通讯及处理的程度测试设计[J].计算机技术与发展,2011,21(6):189-192.
- [7] 阳宪惠.现场总线技术及其应用[M].北京:清华大学出版社,1999:3-17.
- [8] Philips Semiconductors. P8XC591 Single-chip 8-bit microcontroller with CAN controller[S]. 1999.
- [9] 周永学.电子鼓手的计算机控制系统实现[J].科学技术与工程,2007,7(13):3114-3117.
- [10] 褚 杰,原 亮,赵 强,等.基于 TMR-EHW 的高可靠性电机控制系统[J].计算机工程,2009(23):7-9.
- [11] 微软公司. Microsoft Windows CE Communication Guide[M]. [s. l.]: Microsoft Press, 1998.
- [12] 汪 兵,李存斌.EVC 高级编程及其应用开发[M].北京:中国水利水电出版社,2005.