

# 核心交换机嵌入式重启装置研究与设计

刘振海 时间:2011年01月17日 来源:电子技术应用2010年第11期

关键词:[SNMP交换机嵌入式重启ARM920TSP3232Linux](#)  
[Sipex Corporation](#)

**摘 要:** 介绍了[SNMP](#)协议及获取核心[交换机](#)信息的方法,给出了具体的系统设计方案。应用ARM+[Linux](#)构成[嵌入式](#)系统,通过带外管理方法,实现了在核心交换机死机等异常情况下自动对其进行[重启](#)的功能,大大减轻了网络管理员的负担。

**关键词:** SNMP; 交换机; 嵌入式; 重启

交换机是局域网中用来交换、传输数据信息的枢纽,其工作状态的好坏直接关联着局域网网络的数据传输稳定性。特别是作为整个局域网中心的核心交换机,一旦出现问题,则意味着整个网络瘫痪。因此,保障核心交换机的稳定运行是网络管理的一项重要工作。

实践表明,核心交换机出现硬件故障的概率极少,但交换机在长时间工作过程中,很容易出现一些软件故障,如死机、缓存溢出、交换机端口“假死”等情况,解决这类问题的简单方法就是重启机器。所以重启交换机是网络管理员经常做的工作。而管理员的办公地点有时距离交换机很远,而且很多核心交换机没有开关,只能通过拔插电源的方法进行重启,这种方法对机器会造成不良影响。通过超级终端与交换机连接,用命令进行重启,也不太方便。

虽然远程监控已很普遍,但远程监控基于网络和交换机端口,一旦网络堵塞或交换机死机就无法实现信息的监测和控制。而基于SNMP的嵌入式重启装置可以解决这一问题,它不需人工干预,便可在特殊情况下自动实现对交换机的重启工作。

## 1 设计思路

所设计的嵌入式系统应具备:RJ-45口和RS-232接口,RJ-45口与交换机的一个Ethernet口相连,通过SNMP协议获取交换机管理对象库MIB信息[1],检测交换机是否正常;RS-232接口与交换机的Console口相连,通过带外管理方式实现异常情况下对交换机重启功能。

## 2 硬件实现

嵌入式装置的硬件结构框图如图1所示。

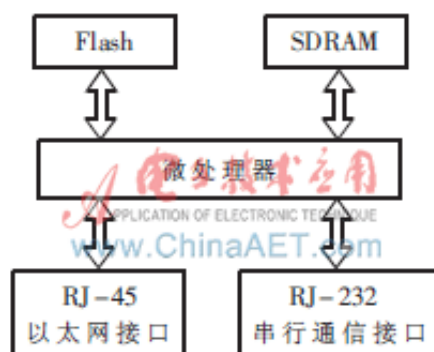


图1 嵌入式装置硬件结构框图

### 2.1 微处理器选择

传统的8位/16位微控制器由于速度慢、功耗大,并且实现网络协议困难,已越来越不能满足高速发展的网络管理需要,随着ARM(Advanced RISC Machine)RISC处理器的不断成熟和嵌入式Linux的不断完善,ARM+Linux嵌入式解决方案日益流行,因此,本系统微处理器选用[ARM920T](#)。ARM920T具有5级指令流水线,采用哈佛结构,具有16/32位RISC体系结构和ARM指令集,处理速度为1.1 MIPS/MHz,内置MMU、独立的16 KB数据Cache、16 KB指令Cache和高速AMBA(Advanced Microcontroller Bus Architecture)总线接口等功能[2]。

### 2.2 网卡接口设计

该系统需要连接交换机以太网端口,通过SNMP协议获取交换机管理对象库MIB信息。系统采用DAVICOM公司的DM9000AEP作为以太网MAC控制器与处理接口。该器件具有10 Mb/s和100 Mb/s的自适应物理层收发器,支持802.3x全双工控制流标准,具有4 KB的双字SRAM,可以减

轻CPU的负担,因此使系统更稳定,处理和传输数据速度更快。RJ-45插座采用HanRun公司的内置网络变压器、状态显示灯和电阻网络的HR911105A,具有信号耦合、电气隔离、阻抗匹配、抑制干扰等优点。

ARM920T网络接口设计电路如图2所示。DM9000AEP芯片的SD0~SD15为16根数据线,与ARM920T数据线相连;命令选择引脚CMD为高电平,访问数据端口,CMD为低电平,访问地址端口;IOR#为读信号引脚,与ARM920T的读信号LnOE引脚相连;IOW#为写信号引脚,与ARM920T的写信号LnWE引脚相连;CS#为片选信号,与ARM920T的片选信号nGCS3引脚相连;INT为中断请求信号,高电平有效,与ARM920T的IRQ-LAN相连;发送端TX+、TX-和接收端RX+、RX-分别连接到HR911105A的发送端和接收端。

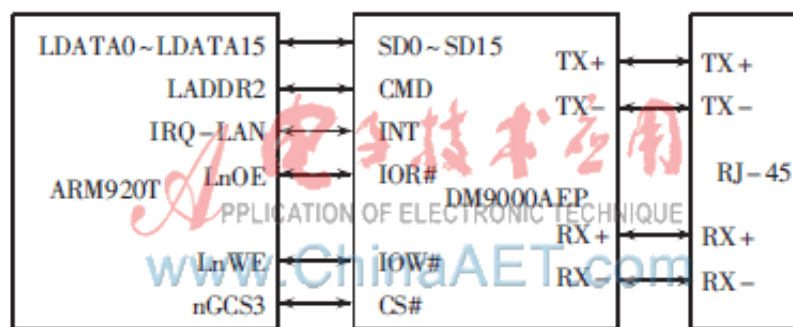


图2 网络接口设计电路

### 2.3 串口接口设计

该系统用带外管理方式对交换机实现重启操作,需要RS-232接口与交换机Console口相连。ARM920T本身就具有串行通信接口,只需将TTL电平转化为RS-232电平即可。Sipex公司的SP3232芯片可以实现这一功能,而且SP3232芯片所需的供电电压低,适合嵌入式系统应用,其外围电路连接简单,仅需几个0.1  $\mu$ F的电容器即可。串口接口设计如图3所示。图3中TXD表示接收数据,RXD表示发送数据,RTS表示请求发送,CTS表示清除发送。当RTS请求发送数据时,需经CTS信号检测,只有CTS信号允许才能发送数据。

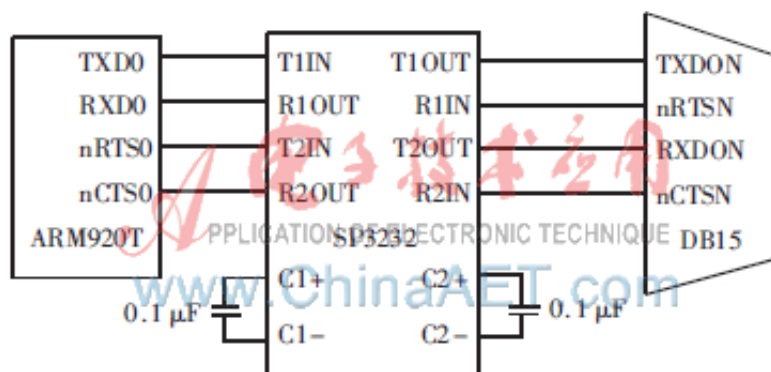


图3 串口接口设计

### 2.4 存储器设计

存储器由SDRAM和Flash构成。SDRAM存储器用来存放操作系统(从Flash解压缩拷入)以及存放各类动态数据。设计中系统采用SAMSUNG公司的K4S561632,容量为32 MB。用2片K4S561632实现位扩展,使数据总线宽度达到32 bit,总容量达到64 MB,将其地址空间映射在ARM920T的bank6;Flash存储器有NOR和NAND两种,用来存放Linux引导加载程序(BootLoader)。NOR Flash的特点是芯片内执行(Execute In Place),即程序可直接在Flash闪存内运行,不必把代码读到系统RAM中。NOR Flash的传输效率很高,在1~4 MB的小容量时具有很高的成本效益,但是很低的写入和擦除速度大大影响了它的性能。NAND Flash结构能提供极高的单元密度,可以达到高存储密度,并且写入和擦除的速度也很快,应用NAND Flash的困难在于Flash的管理和需要特殊的系统接口。本系统采用Intel公司的NOR Flash 28F640芯片,其容量为8 MB,将其地址空间映射在ARM920T的bank0。

## 3 软件实现

### 3.1 操作系统

ARM芯片获得了许多实时操作系统供应商的支持,有: Windows CE、Linux、Palm OS、VxWorks、pSOS、Nucleus等。因Linux源代码的公开特性, Linux系统的应用越来越广泛。将Linux应用于ARM芯片, 可以采用一些现成的针对ARM芯片的嵌入式Linux系统, 也可根据需要自行对Linux内核进行裁剪。本系统采用自行裁剪的方法, 使用make menuconfig命令对Linux内核进行配置, 整个配置以维持系统运行的最小化为原则。需要注意的是内核应支持proc、ext2文件系统、UNIX domain sockets、TCP/IP networking、UART等选项, 以保证系统能实现RJ-45口和RS-232接口对外通信。实现这些功能的内核静态压缩映像大约为500 KB[3]。为了实现系统功能, 还需编写相应的网络接口和串口驱动程序, 建立根文件系统。

### 3.2 引导加载程序

操作系统的启动离不开引导程序, Linux引导加载程序(BootLoader)由NOR Flash的地址0x0处开始执行, 完成存储设备初始化、堆栈初始化、用户环境初始化等, 最终BootLoader把操作系统内核映像加载到RAM中。本系统引导加载程序采用ARM-BOOT修改而成。

### 3.3 主程序

主程序每次在设备开机后自动运行, 在交换机上先配置一个VLAN接口用于与嵌入式设备通信。嵌入式设备的网络接口IP地址应与交换机的这个VLAN接口IP处于同一网段。为避免与常用私网地址冲突, 可以将VLAN接口IP设为192.168.255.1, 将嵌入式设备的网络接口IP地址设为192.168.255.2。定时器时间设为1 min, 每隔1 min对交换机发送1次SNMP报文, 获取交换机状态进行分析, 如连续3次(3 min)发现通信失败或判定交换机状态异常, 则对交换机进行重启。主程序流程图如图4所示。

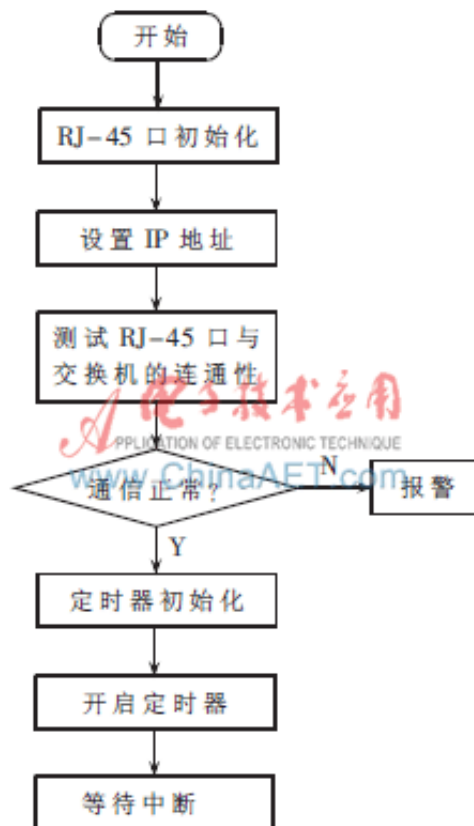


图4 主程序流程图

### 3.4 交换机状态监测程序

简单网络管理协议(SNMP)是Internet组织用来管理Internet的网络协议, 其定义了传送管理信息的协议消息格式及管理者和设备代理相互之间进行消息传送的规程。在SNMP管理模型中有三个基本组成部分: 管理者(Manager), 被管代理(Agent)和管理信息库(MIB)。管理者的管理工作是通过轮询代理来完成的, 管理者可以通过SNMP 操作直接与被管代理通信, 获得即时的设备信息, 对网络设备进行远程配置管理或者操作, 也可以通过对数据库的访问获得网络设备的历史信息, 以决定网络配置变化等操作。SNMP被管代理是指用于跟踪监测被管理设备状态的特殊软件或硬件, 每个代理都拥有自己本地的MIB。MIB信息即被管资源, 而网络管理中的资源是以对象表示的, 每个对象表示被管资源的某方面属性, 这些对象形成了MIB库。MIB是一个树形结构, SNMP协议消息通过遍历MIB树形目录中的节点来访问网络中的设备。网络管理者通过

对MIB库的存取访问实现配置管理、性能管理、故障管理、安全管理、计费管理等五大管理功能。SNMP提供有三类操作: Get、Set和Trap。Get操作实现对被管理对象所表示的管理信息的读操作，Set操作实现对被管理对象的管理信息进行写操作，Trap 操作实现被管理设备向管理工作站报告设备出现的异常事件。

交换机状态异常或死机的原因主要有：散热不好致温度过高；病毒攻击；交换机背板带宽过低；负荷过大等。交换机状态异常或死机最终表现为: CPU利用率和内存利用率过高、端口丢包率过大、与交换机端口连接不通等。因此，需要对交换机的CPU利用率、内存利用率、端口丢包率等进行监测，同时依据与交换机的SNMP通信是否正常来判断交换机是否出现异常或死机。本系统通过SNMP协议与交换机的SNMP被管代理通信，读取(Get)相应MIB节点数据，获知CPU利用率、内存利用率、端口丢包率等信息。

下面以华为3com公司的S7500系列交换机为例，说明本系统需要使用的MIB节点, 如表1所示。

表 1 系统需要使用的 MIB 节点

监测	1.3.6.1.4.1.2011.6.1.1.1.2	hwCpuCostRate
CPU	1.3.6.1.4.1.2011.6.1.1.1.3	hwCpuCostRatePer1Min
利用率	1.3.6.1.4.1.2011.6.1.1.1.4	hwCpuCostRatePer5Min
监测	1.3.6.1.4.1.2011.6.1.2.1.1.2	hwMemSize
memory	1.3.6.1.4.1.2011.6.1.2.1.1.3	hwMemFree
利用率		
监测	1.3.6.1.2.1.2.2.1.13	ifInDiscards
端口的	1.3.6.1.2.1.2.2.1.11	ifInUcastPkts
丢包数	1.3.6.1.2.1.2.2.1.12	ifInNUcastPkts

状态监测程序的流程图如图5所示。



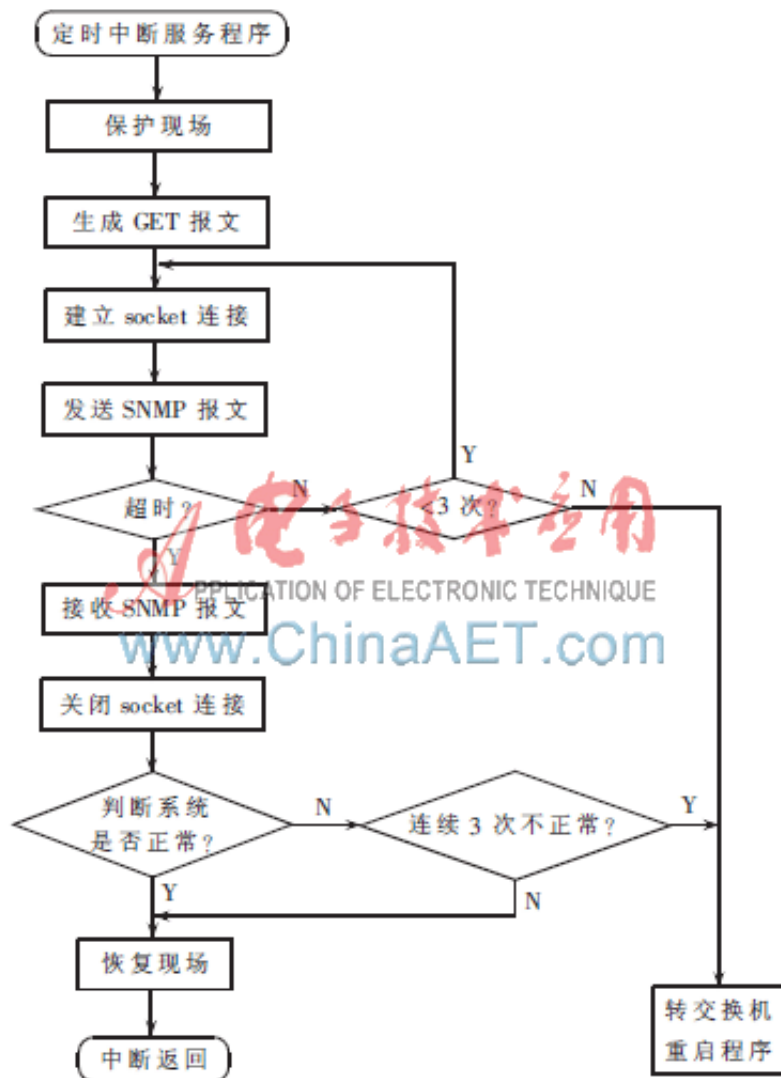


图 5 交换机状态监测程序流程图

### 3.5 交换机重启程序

远程监控一般是通过网络经交换机以太网口对交换机实行控制, 大多数情况下是可行的, 但在交换机死机时, 所有端口都不通, 无法通过以太网口对交换机发送命令, 对交换机实行带内管理。对这种特殊情况, 带外管理是唯一可行的方法。

带外管理是通过RS232总线与交换机的Console口进行串行通信, 给交换机下达命令。一般波特率为9 600 b/s, 无流控, 每帧包括8位数据、1位起始位、1位停止位共计10 bit, 无奇偶校验位。

需要注意的是, 不同交换机要实现重启, 发送的指令字符可能不一样, 要根据具体交换机型号而定。以华为3com公司的S7500系列交换机为例, 重启程序的流程图如图6所示。



图 6 交换机重启程序流程图

本嵌入式系统实现了在核心交换机死机等特殊情况下自动对其进行重启的功能，大大减轻了网络管理员的负担。将本系统进行扩展，可以根据需要实现对交换机特定方面的本地监测和控制功能。特别是该系统将来可以与核心交换机融为一体，有效增强核心交换设备的智能化和自我修复能力。

#### 参考文献

- [1] MCCLOGHRIE K, ROSE M. Management information base for network management of TCP/IP-based internets [DB/OL] [M]. <http://www.rfc-editor.org/rfc/rfc1066.txt>; August 1988.
- [2] 曲鹏, 严胜刚, 李博丞. 基于嵌入式Web服务器的智能家居主控制系统[J/OL]. <http://www.21ic.com/app/embed/200907/45684.htm>. 2009-07-30.
- [3] 刘振海. 基于软盘的Linux系统的制作[J]. 微计算机应用, 2004, 25(6): 700-710.