

嵌入式处理器 8240 的板级支持包的设计与实现

冯先成,司擎华,李小鹏,张铁男

(武汉工程大学电气信息学院,湖北 武汉 430074)

摘 要:光纤、无线网络通信的嵌入式操作系统平台与网络应用程序的开发、测试、应用非常广泛,为满足网络交换对软件可靠性的要求,提出一种可靠的板级支持包设计方案,采用处理器 8240、伽利略 48304 交换芯片的最小化系统和 VxWorks 操作系统内核,基于上电初始化开始的板级支持包设计流程,建立 Vxworks 系统的板级支持包调试平台,实现板级支持包源代码的设计、移植烧制与测试.实验结果表明通过文件传送协议将修改、编译后的板级支持包成功下载到目标板的闪存中,实现可靠稳定的远程登陆和网络管理.

关键词:板级支持包;嵌入式系统;网络交换;制备过程

中图分类号:TP311

文献标识码:A

doi:10.3969/j.issn.1674-2869.2012.11.012

0 引 言

VxWorks 操作系统主要有板级支持包(Board Support Package,以下简称:BSP)、微内核(wind)、网络系统、文件系统、输入输出(input/output, I/O)系统^[1].BSP 是指支持一个目标板的硬件系统正常运行的与硬件密切相关的函数集合,主要包括目标系统启动时的硬件初始化例程和目标板上控制各个硬件设备正常运行的驱动程序,还要提供 VxWorks 与硬件环境之间的主要接口.根据目标板的硬件构成,BSP 设置各种与硬件参数有关的宏,设置内存映射表,编写和修改相应的设备驱动程序,连接中断服务程序,根据上层和系统的需要生成相应的镜像(image)文件,创建相应的任务,并启动应用程序^[2].

1 硬件系统

网络交换系统的主控交换的 BSP 包括两大部分:Motorola 嵌入式处理器 8240(MPC8240)最小系统和相关外围设备的驱动.

MPC8240 主控交换板最小系统包括:中央处理器(CPU)、同步动态随机存储器(SDRAM)和闪存(Flash),闪存包括用于系统自启动的根配置闪存(BOOT FLASH)文件和用于程序、数据存储的程序闪存文件两部分、串口驱动程序,82559 网口驱动程序,外设部件互连接口 9030(PCI9030)接口控制器、调试接口及部分控制逻辑等.最小系统

BSP 的目标就是通过配置最小系统使之能够完成配置过程,并可以使用网口和串口进行上层软件的调试.

主控板主要由下列 a—g 共 7 个部分组成,主控板结构框图如图 1.

a. Motorola 的 8240CPU. 8240 集成了一个 PCI 桥—MPC107,一个嵌入式可编程中断控制器 EPIC,一个存储器直接访问 DMA 控制器及 I2C 控制器等.

b. Intel 82559 10M/100M 自适应以太网芯片.

c. Galileo 48304 交换芯片.

d. 16552 串口芯片.

e. HDLC 芯片.

f. 一片 SST 39VF040 存放 Bootrom.

g. 四片 ATMEL: AT49B1614AT—90TI 用于存放版本和 TMS 的数据.

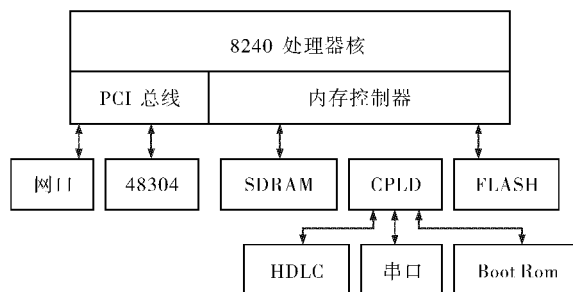


图 1 主控板结构框图

Fig. 1 Structure of main control board

收稿日期:2012-3-23

基金项目:武汉工程大学研究生教育创新基金项目(NO:201161);武汉工程大学大学生校长基金

作者简介:冯先成(1968-),男,安徽庐江人,副教授.研究方向:综合宽带光纤接入系统、家庭网络、嵌入式技术.

2 BSP 处理流程、移植与烧制

2.1 BSP 处理流程

设计、调试 BSP 软件,首先分析整个操作系统从上电初始化开始的执行流程,BSP 执行处理流程如图 2 所示^[3-4].

当系统上电后,MPC8240 的 CPU 会在 0x02800100 执行,这段地址空间物理上是对应于一块配置只读内存,其中保存着操作系统的引导代码,VxWorks 按图 2 所示的流程引导过程如下:

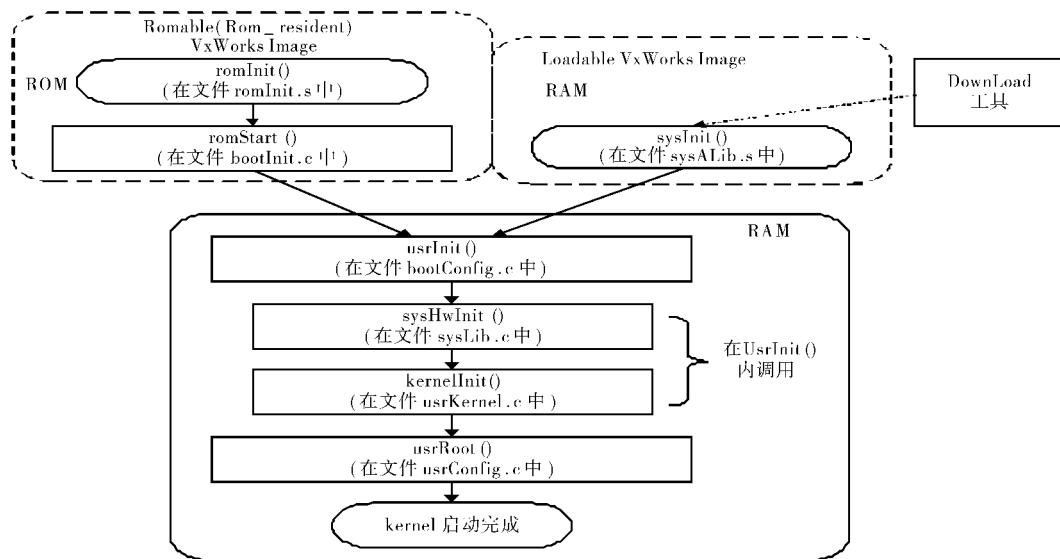


图 2 BSP 处理流程

Fig. 2 Processing flow of BSP

c. 内核初始化:所有的板级硬件初始化完成后,就开始操作系统核心代码的加载,由其建立多任务环境,创建根任务 usrRoot 文件;

d. 输入输出系统初始化:在根任务中,继续完成输入输出设备,网络协议栈及设备驱动的加载工作。

在 VxWorks 引导成功后,它会在根任务 usrRoot 中调用 usrAppinit 函数,用户可以在这个函数中创建自己的任务或者启动相应的软件模块。

2.2 BSP 移植

2.2.1 修改 TGT_DIR \ src \ drv \ pci \ pciConfigLib.c 程序

a. 保留原来所有的程序不动

b. 增加一个函数如下,该函数可以找出一台 PC 上同种类型的多个网卡,而原来的函数 (pciFindDevice) 只能找出一个网卡

a. CPU 初始化:文件 romInit 位于 CPU 的复位向量中,一个向量一般最多为 256 字节,因此不能放太多的程序在此位置. 这里仅仅完成如下工作^[6]:

①关闭中断,初始化 CPU 的一些基本寄存器

②配置 SDRAM,使其可以使用

③初始化堆栈指针,调用 romStart 文件

b. 其它初始化:包括内存初始化、CACHE 初始化、BSS 初始化、硬件初始化. CPU 的一些初始化是在 sysInit 函数中完成的. 系统真正的板级硬件初始化是在 sysHwInit 函数中完成的;

STATUS pciFindDeviceArray

(int vendorId, /* vendor ID */

int deviceId, /* device ID */

int index, /* desired instance of device */

int * pBusNo, /* bus number */

int * pDeviceNo, /* device number */

int * pFuncNo, /* function number */

int * pDeviceNum /* The Card Device Number Chen. AiMin */

)^[6]

2.2.2 修改 TGT_DIR \ config \ pcPentium \ sysEl3c90xEnd.c 程序

a. 修改 sysEl3c90xPciInit 函数,增加对同种类型的多块网卡进行初始化,原来的只能对一块 3C905 网卡进行初始化;

b. 修改 sysEl3c90xEndLoad 函数,原来该函数只是默认第一个网卡,现支持多个 3C905 网卡;

c. 该函数前面的 LAN_CARD_NUM 宏定义了该 PC 有最多可以有几块 3C905 网卡,现在定义成 3,最好不要修改。

2.2.3 修改 Config. h 定义 BSP VERSION、default—boot—line 的 BOOT 参数、Console—Baud—Rate、RAM—HIGH—ADRS、RAM—HIGH—ADRS、ROM—BASE—ADRS、ROM—TEXT—ADRS。

```
#define RAM_HIGH_ADRS 0x00450000
/* RAM address for ROM boot
#define RAM_LOW_ADRS 0x00100000
/* RAM address for kernel */
```

2.2.4 修改 MAKEFILE 为了便于对 BOOTROM 中 bootConfig. c 进行修改,可以把 bootConfig. c 从 all 目录拷入相应的 BSP 目录 (pcPentium 目录),修改该目录下的 MAKEFILE,增加如下—行:BOOTCONFIG = bootConfig. c

修改后的 BSP 运行结果:

```
CPU: EST est8240-MPC8240(603e)
Version: 5.4
BSP version: 1.2/2
Creation date: Jan 14 2012, 19:56:33
Press any key to stop auto-boot...
6
[Boot]: p
boot device: dcf
unit number: 0
processor number: 0
host name: wx
file name: vxWorks
inet on ethernet (e): 10.21.14.56
host inet (h): 10.21.14.188
gateway inet (g): 10.21.1.254
user (u): wx
ftp password (pw): 3592
flags (f): 0x0
target name (tn): est8240
```

2.3 BOOTROM 的烧制

a. 根据 BSP 生成 BOOTROM,注意生成时选择 TORNADO→Build→Build Boot Rom 选择 BSP,注意选择生成 Bootrom_uncmp. hex.

b. 打开 VisionClick,将主机、仿真器和 CPU 板连通,在 Terminal 窗口下输入命令 in,将仿真器中配置好的 CPU 相关寄存器的值下载到板子上去。

c. 下载 bootrom_burn_amd. bdx 到目标板,这段代码存放在 SDRAM 中的 0x01800100 开始

的一段地址中(如果烧写 SST39VF040 的 FLASH 芯片,则使用 bootrom_burn_sst. bdx)。

d. 将生成好的 Bootrom_uncmp. hex 在 VisionClick 中转化为 bootrom_uncmp. bdx,下载 bootrom_uncmp. bdx 到目标板。

e. 输入命令以运行烧写 bootrom 程序代码到 flash 中去,方法为在 Terminal 窗口下输入命令 go 01800100。

f. 用仿真器察看内存 0x01c200a0 地址域会看到进度指示,包括操作时当前写操作地址、失败计数。

g. 最后直到在 0x01c200b0 显示 DONE 表示成功,FAIL 表示失败,整个过程大约 30 s 左右。

h. 点击 VisionClick→Run→Halt execution,或者直接点击红色按钮 Holt Target 停止板子上程序的运行。

i. 重新加电或者按复位键,打开超级终端则可以看到引导信息的输出。

3 实验结果

实验步骤如下:

a. 首先按照 2.2 节修改 BSP 文件。

b. 编译 BSP 文件,生成 VxWorks 镜像文件^[7],如图 3 所示。

步骤 1: 备份并编辑.../_BSP_WDB/config. h (可选),在 config. h 中找到宏定义

```
#define DEFAULT_BOOT_LINE "lsi(0,0)
host:vxWorks h=192.168.0.113 e=192.168.0.
127 u=my pw=you tn=l2sw24 f=0x00"
```

根据需要修改参数:

h = 主机 IP 地址

e = 目标机 IP 地址

u = ftp server 用户

pw = ftp server 用户口令

tn = 目标板网络机器名

说明:一般不修改,可在目标机开机时,通过引导提示行,用户重新配置

步骤 2: 进入 DOS 命令行方式,进入 .../_BSP_WDB

步骤 3: 执行 make bootrom. hex. 生成 VxWorks 镜像文件 bootrom. hex

c. 将镜像文件通过 FTP 文件传送协议下载到目标板的闪存中,把 HOST 主机的串口和目标板的串口连接起来,HOST 的以太网口和目标板的调试以太网口连接起来。打开“开始→程序→TORNADO→VxWORKS COM1 (超级终端)和

FTP server”,在 FTP server 的 Securiry→users/rights,出现如图 4 对话框,点击 **New User...**,填入用户名和密码,再在 **Home Directory** 栏填入,生成 VXWORKS

映像的位置,按 **Done** 按钮。把目标板上电,过一会在超级终端打印出如图 4 信息^[8]。

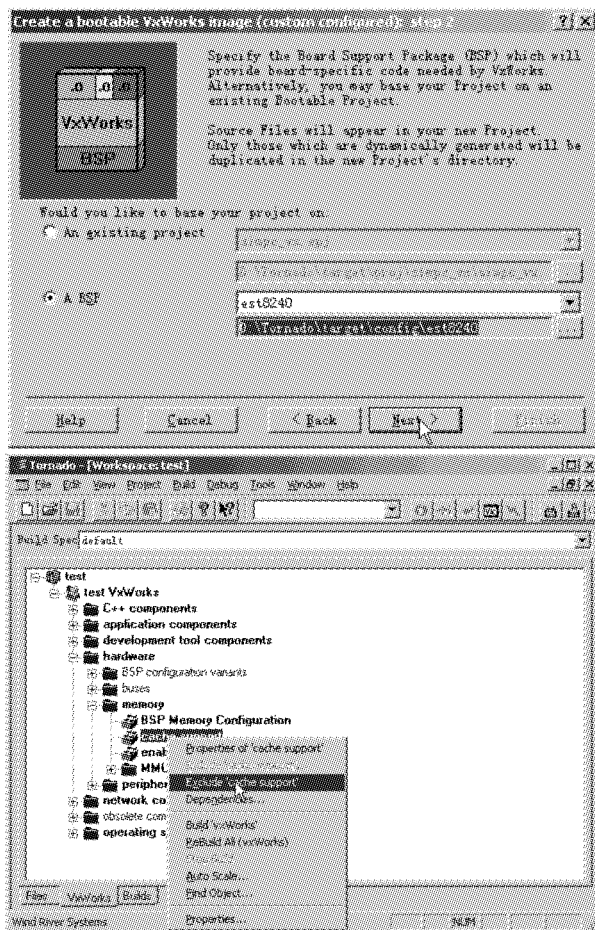


图 3 8240BSP 的安装、VxWorks 镜像文件的生成

Fig. 3 Installation of 8240BSP, VxWorks file



图 4 通过 FTP 下载镜像文件到目标板

Fig. 4 Download the image file to target board with FTP

d. 通过主机的 Telenet 命令远程登陆目标,进行调试.输入 help,可以查看命令行信息,如图 5.用‘P’命令查看 BOOT 参数,超级终端打印出目标板 BOOT ROM 的参数配置,在命令提示符下输入“c”命令,把 HOST 的 IP 和网关修改为所用 HOST 的,把 FILE NAME 栏改为 VXWORKS,把 USER 和 FTP PASSWORD 改为 FTP SERVER 刚才设置的.输入后,下载成功,WDB(WIND RIVER 调试)准备好了。

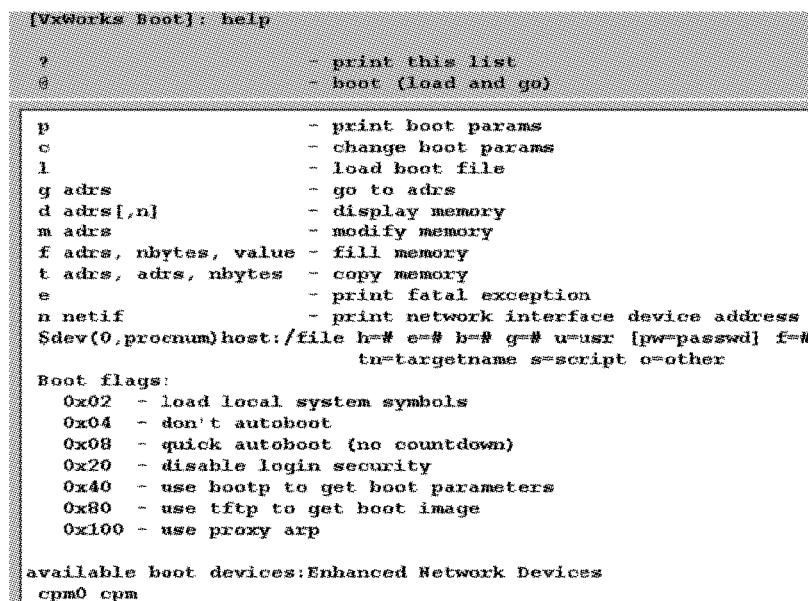


图 5 目标板调试

Fig. 5 Test of target board

4 结 语

MPC8240 是通信领域嵌入式产品使用较多的芯片,针对通信领域该类芯片的最小化系统,笔者基于 MPC8240 的最小化系统提出一种可靠的 BSP 设计流程方案,完成 BSP 源代码的设计、移植烧制与测试。测试证明本方案有较好的可靠性和良好的可操作性,对于 Vxworks 上层应用软件和底层硬件驱动的并行开发有着很重要的意义。

参考文献:

- [1] 李明,胡立德,尹安治. 军用嵌入式系统的可靠性设计[J]. 电子元器件应用,2007,9(10):71-72,75.
- [2] 冯先成,李寒,张铁男. 基于 MPC850 VxWorks 系统的 BSP 设计[J]. 计算机与数字工程,2012,40(6):39-43,75.
- [3] 蒋鲲鹏,芦东昕,缪敬. Linux 与 VxWorks 的板级支持包开发的比较与分析[J]. 计算机工程,2003,29(22):51-53.
- [4] 隋霞,许录平. 基于 VxWorks 的 BSP 技术分析[J]. 微计算机信息,2006,22(23):86-88.
- [5] 王韬,印勇,刘国金. 基于 VxWorks 的嵌入式系统实验教学设计[J]. 实验室研究与探索,2007,26(1):52-54,105.
- [6] 孙海彬. 支持 PCI 总线的 CORBA 可插拔协议研究与实现[D]. 长沙:国防科技大学,2010.
- [7] 王韬,杨士中,谭晓衡. 基于 MPC860 和 VxWorks 的嵌入式中断处理设计[J]. 电讯技术,2005,45(1):45-50.
- [8] 王柯. 基于 Pentium4 处理器的 VxWorks 板级支持包设计[J]. 电子科技,2011,24(6):57-59.

Design and implementation of board support package on MPC8240

FENG Xian-cheng, SI Qing-hua, LI Xiao-peng, ZHANG Tie-nan

(School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan 430074, China)

Abstract: The platform, web application development, testing, application of optical fiber, wireless network communication on embedded operating system have wide applications. In order to meet the requirements of network switching on software reliability requirements, one kind of reliable design of board support package(BSP) was proposed, the minimum system of MPC 8240 and Galileo 48304 based on VxWorks operating system kernel was adopted; based on design process of BSP which started after the power-up initialization, debugging platform of BSP on Vxworks system was established, source code design, transplantation and test of BSP was realized. Experimental results show that the modified compiled BSP is successfully downloaded to the target board flash by file transfer protocol, and the remote login and network management is stable and reliable.

Key words: board support package; embedded system; network switching; preparation process

本文编辑:苗 变