

# 基于 MPC850 VxWorks 系统的 BSP 设计\*

冯先成 李 寒 张铁男  
(武汉工程大学 武汉 430074)

**摘 要** 首先介绍嵌入式系统 VxWorks 及其启动过程;分析了基于 MPC 850 最小系统的硬件和 BSP 的执行流程;重点研究了 MPC850 BSP 的设计与应用程序加载,包括其硬件驱动程序的设计、应用程序加载、建立调试环境和 target server 的配置等;基于 MPC850 系统的 Vxworks BSP 调试平台的实现,对 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。

**关键词** VxWorks; MPC850; 板级支持包; 目标服务器  
**中图分类号** TP301

## Design of BSP Based on VxWorks System of MPC850

FENG Xiancheng LI Han ZHANG Tienan  
(Wuhan Institute of Technology, Wuhan 430074)

**Abstract** First the embedded system VxWorks and starting process are introduced. Based on MPC 850 the system of hardware and BSP execution flow are analyzed. The MPC 850 design of BSP and application program loading are focused on, including its hardware driver design, application program loading, build debugging environment and target server configuration. Realization of Vxworks BSP debug platform on MPC850 system has a very important significance with the VxWorks application software and hardware driver parallel developing.

**Key Words** VxWorks, MPC850, board support package, target server  
**Class Number** TP301

### 1 引言

用户程序
操作系统
BSP
硬件

VxWorks 是专门为实时嵌入式系统设计开发的操作系统内核,为程序员提供了高效的实时多任务调度、中断管理,实时的系统资源以及实时的任务间通信<sup>[1~2]</sup>。基于 VxWorks 操作系统的应用程序可以在不同 CPU 平台上轻松移植<sup>[3]</sup>。BSP 是介于底层硬件和上层软件之间的底层软件开发包,它将系统中与硬件直接相关的一层软件独立出来。BSP 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,因此它不是普通的硬件驱动程序,如图 1 所示<sup>[4]</sup>。

图 1 BSP 的位置

开发 BSP 主要完成的工作:根据目标板的硬件构成,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 image,创建相应的任务,并启动应用程序。

### 2 Vxworks 映像分类及其启动过程

VxWorks 在启动过程中涉及到两个映像文件:

bootrom 映像和 VxWorks 映像。bootrom 文件是从 ROM

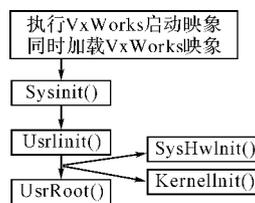


图 2 VxWorks 加载映像流程

引导 VxWorks 的引导目标模块,用于对目标板进行初始化,加载 VxWorks 操作系统映像。VxWorks image 包含 VxWorks OS,最终运行于目标板上<sup>[5~6]</sup>。VxWorks 映像按加载类型

可分为:可下载映像和可引导映像;按运行方式分为在 ROM 中和在 RAM 中两种。图 2 是 VxWorks 加载映像流程。可下载映像(Loadable Image),它实际包括两部分,一是 VxWorks,二是 boot ROM,两部分是独立创建的。可引导映像(Bootable Image)是将引导程序和 VxWorks 融为一体的映像,包括不驻留 ROM 的映像和驻留 ROM 的映像两种类型。

### 3 MPC850

#### 3.1 850CPU

MPC850 具有一个带有内存管理单元(MMU)和指令数据 Cache,32 位的 PowerPC 结构的 CPU。并内嵌有 CPM(Communication Processor Module),其功能及结构框图如图 3<sup>[7]</sup>。

\* 收稿日期:2011 年 12 月 8 日,修回日期:2012 年 1 月 18 日  
基金项目:武汉工程大学研究生教育创新基金项目(编号:No. 61)资助。  
作者简介:冯先成,男,硕士,副教授,研究方向:计算机网络通信。

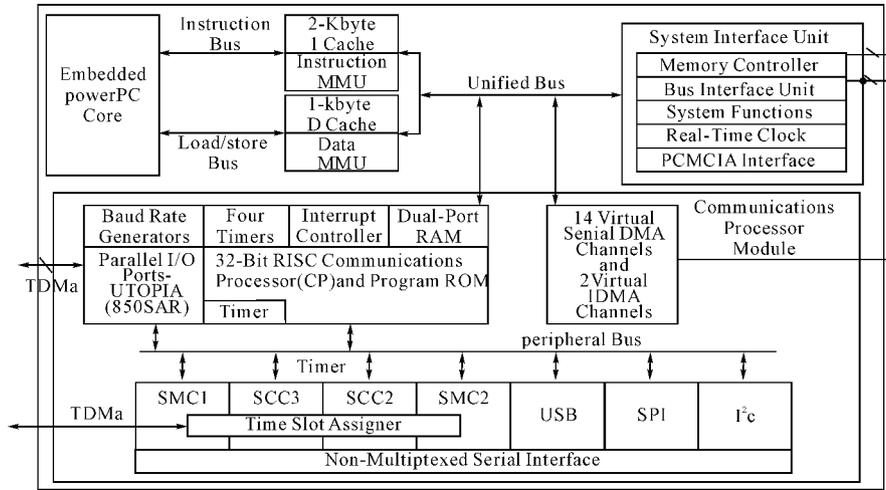


图 3 MPC850 的内部结构图

- 32 位的 PowerPC 核心及所带的 MMU 及指令,数据 Cache。在 40MHz 时钟时为 52MIPS 的指令速度,在 50MHz 时钟时为 66MIPS 的指令速度

- 系统接口单元(SIU)
- 通讯处理器模块(CPM)

### 3.2 MPC 850 最小系统

典型的 MPC 850 最小

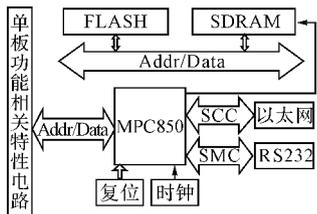


图 4 MPC850 的单板电路框图

系统用于程序启动和加载的 BOOT FLASH,用于存储数据及版本的 FLASH 和用于暂存数据的 SDRAM;由一个或两个 SCC 口构成的以太网口用于系统内部通讯,一个 SMC 口构成的 RS232 用于

调试。同时每个单板上都有一个 BDM 调试接口,另外还有复位和时钟电路。如图 4 是 MPC 850 最小系统的电路框图。由 MPC850CPU、Flash、SDRAM、串口、网口、复位、初始配置、系统时钟等组成。通过仿真器 VisionProbeII 把程序下载到 SDRAM 中,然后再通过仿真器把程序下载到 FLASH 中,FLASH 主要是存储程序或 Vxworks 操作系统。当上电时,通过上电复位,配置 MPC850CPU 的初始配置。然后把存储在 FLASH 中的程序或 Vxworks 操作系统,复制到 SDRAM 中,这时只要 MPC850 CPU 运行起来,可以让 MPC850 CPU 在 SDRAM 中执行各种用户或系统的需求功能。通过标准 RS232 接口完成对 MPC850 CPU 进行控制。单元电路的功能:

• CPU:CPU 芯片 MPC850 对各电路进行操作和控制,实现各项功能。

• 电源单元:MPC850 需要的电压为 3.3V,整板供电电压 3.3V。

• 串口:MPC850 自带串口。经过 MAX3221 做 RS232 线路驱动。串口的主要功能就是数据的串 / 并转换,并执行 RS232 口数据的输入 / 输出操作。

• 网口:网口是由变压器 23Z467SM、LXT905 和 MPC 850 中的 SCC 2 中的以太网控制器组成。其中

LXT905 相当于 PHY 层,MPC850 中的 SCC 2 中的以太网控制相当于 MAC 层。双绞线来的信号,经过变压器进入到 LXT905 中进行曼切斯特解码,再进入到 SCC 2 中的以太网控制器,然后通过该控制器把以太网帧存储在缓冲区

内。  
• SDRAM:MPC850 提供对 SDRAM 的支持,能够自动完成动态存储器的刷新等操作。支持多种方式的行列地址宽度;支持 8,16,32BIT 数据线宽。两块 SDRAM,大小共为 32MByte。

### 3.3 MPC850 BSP 的执行流程

底层控制软件由 windKernel、BSP 组成。windKernel 是实时系统的内核,BSP 则是整个 CPU 盘的硬件初始化包,它完成 MPC850 的 Microprocessor、Hardware Interface 和 Communications Processor 的初始化,网口和串口的收发控制程序等组成,它们结合在一起生成 BOOTROM,存放在 CPU 盘上的 AM29LV017D 或 SST39VF016(FLASH)芯片中。当 CPU 盘加电时直接跳到上面运行这一文件<sup>[9~10]</sup>。

• WindKernel 是这个实时系统的内核,是别的程序运行的基础,由于软件并不提供它的源码,我们并不需要对它进行开发,只是在硬件初始化时运行这个内核即可。

• MPC850 的 Microprocessor、Hardware Interface 和 Communications Processor 的初始化程序:完成对 MPC850 的寄存器的配置,完成对通信处理器 SCC2、SMC1、SMC2 的存储空间,寄存器的配置,BD 表的初始化,并挂接相应的中断,同时为它们配置需要的网络协议。MPC850 将部分通信处理器和核心处理器集成在一起,其中的 SCC2 被配置为以太网控制器模式,SMC1、SMC2 被配置为串行处理器模式。

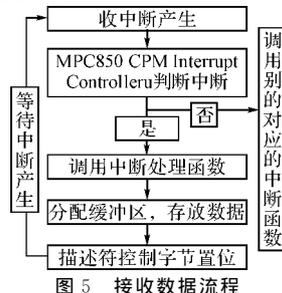


图 5 接收数据流程

• 网口、串口的收发控制程序,其中接收数据的流程如图 5。发送数据是在程序运行过程中需要时,即当接收的数据处理和部程序运行过程中需要通过网口或串口发送数据时,自主调用相应的发函数来出来

完成。

在启动 windKernel,完成了 CPU 盘上硬件初始化之后,BOOTROM 将发起一个任务,来完成从 CPU 盘上的 FLASH 芯片中读取,或者通过 FTP 从连接的 PC 机上下载整个设备的操作系统、应用软件,并且运行这些软件。过程如图 6。

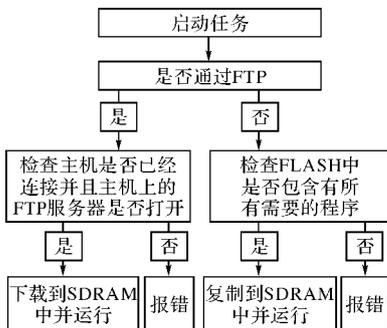


图 6 下载操作系统、应用软件过程

### 4 MPC850 BSP 的设计与应用程序加载

#### 4.1 BSP 设计过程

BSP 开发处于整个嵌入式开发的前期,是后面系统上应用程序能够正常运行的保证。大体上,BSP 的设计与开发可以分为以下几个步骤:

- 1) 建立开发环境。安装 VxWorks 集成开发环境 Tor-nado,采用宿主目标机交叉开发。
- 2) 选择合适的 BSP 模板。一个合适相近的 BSP 模板可以大大减少开发周期。
- 3) bootrom 的运行。修改、添加 WIND 内核激活前的初始化代码。
- 4) VxWorks 的配置。修改或添加所需要组件及驱动程序,激活内核,实现基本操作系统功能。
- 5) 测试与验证。正确实现 bootrom 及 VxWorks 的下载,根据不同需要,进行修改与测试。

结合实际中的工程 VxWorks 主机工具与目标机交互关系如图 7 所示。

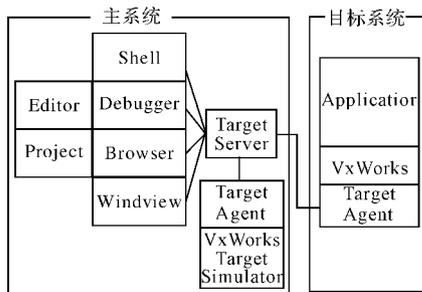


图 7 VxWorks 主机与目标机交互

#### 4.2 MPC850 BSP 设计流程

##### 1) 地址映射

850 共有 8 个片选信号,即可支持 8 个 Memory bank,其中片选 0 为启动片选,即我们常说的 BOOT,850 支持一种 General-purpose chip-select machine(GPCM)和两种 user-programmable machines(UPMs)。

在 VxWorks 中,修改/bspname/ads860.h 文件,内部

存储器修改宏。以下为一段的 bank 配置:

```
SDRAM(2M)0x00000000-0x01FFFFFF
BootFlash(512K)0x02800000-0x0287FFFF
Flash(2M)0x07000000-0x071FFFFFFF
/* CS0: bootflash 512K,8bit,start address 0x02800000 */
#define BR0_VAL 0x02800401 /* Chip select 0 for BOOT memory */
#define OR0_VAL 0xFFFF00974 /* Operation register 0 initial value */
/* CS1: Flash,2M,16bit,0x07000000 */
#define BR1_VAL 0x07000801 /* Chip select 1 for Flash memory */
#define OR1_VAL 0xFFC00954 /* Operation register 1 initial value */
/* CS2: SDRAM,2M,16bit,0x0 */
#define BR2_VAL 0x00000881 /* Chip select 1 for SDRAM 4MB */
#define OR2_VAL 0xFFE00800 /* Operation register 2 initial value */
/* CS3: PHY 8BIT */
#define BR3_VAL 0x04000401
#define OR3_VAL 0xFFFF8970
#define INTERNAL_MEM_MAP_ADDR0x03000000
#define INTERNAL_MEM_MAP_SIZE0x00010000 /* 64 K bytes */
```

##### 2) 频率设置

850 系统可以通过系统锁相环将晶振频率后提供系统时钟,对系统锁相环设定是通过寄存器 PLPRCR 完成。在 PLPRCR 的 MF 比特中设定倍频因子。倍频因子=系统时钟/晶振频率-1;

例如:锁相环要把 4M 倍频成 48M,倍频因子就是 11;操作系统的设定:

```
在 ads86.h 和 config.h 中设定系统频率和倍频因子
#define CRISTAL_FREQ4000000
#define FREQ_50_MHZ50000000 /* 50 Mhz */
#define SPLL_FREQ_REQUESTEDFREQ_50_MHZ /* 50 Mhz */
#define SPLL_MUL_FACTOR ((SPLL_FREQ_REQUESTED / CRISTAL_FREQ) - 1)
```

锁相环的设定在 romInit.s 中完成。

##### 3) 中断

PowerPC 上的中断可以分成两大类:一类由某些系统条件或事件引起的,它是异步的,这就是通常所说的中断;另一类是指令执行引起的,它是同步的,一般称之为异常(exception)。经常处理的中断是外部中断,即 external Interrupt,这一中断是由 SIU 中的中断控制器通过 IREQ 向 PowerPC 内核报告的,它管理了第二级中断,这一级中断共 16 个,按优先级次序为:

```
(IRQ0>ILEVL0>IRQ1>...>IRQ7>ILEVL7) 其中 IRQ0 为非屏蔽中断。
```

VxWorks 中要求 BSP 实现的中断接口有:

- intConnect:将指定的中断服务程序连接于指定的中断向量
- intEnable:使能指定中断向量对应的中断
- intDisable:禁止指定中断向量对应的中断

##### 4) CPM 功能

850 的 CPM 是通过使用 buffer descriptors(BDs)作为和通信 buffer 的接口,BD 位于双口 RAM 内的一端空间内,如果使用多个 BD 就是我们称作 BD 环。

网口和串口是 850 最小系统中的 CPM 功能,在驱动程序中使用 BD 作为控制接口。CPM 在双口 RAM 内有一段参数区,包含了 USB、SCC、SPI、I<sup>2</sup>C 和 IDMA 通道的操作参数。

5) 串口功能

为帮助调试,一般单板都提供一个调试串口;常常用于和 PC 机的超级终端相连,用于程序的调试。如果所使用的串口数小于等于两个,一般直接将 MPC8xx 的 SMC 口作为串口(如果有必要,也可以用 SCC 作为串口,即配置为 UART 模式)。如 MPC860 集成的 SMC,其驱动程序也位于 drv\sio 目录下。一般也不需修改。最多是串口的收发引脚配置需要修改,目前使用的 RS232 串口大都是采用收发和地三根线的连接方式。VxWorks 中对于串口驱动的配置程序部分,位于 BSP 目录的 sysSerial.c 文件中,其中定义了一个参数数组 devParas,用以定义串口的配置参数,如果要增加多个串口,只需在数据中添加多个元素接口。系统在引导过程中,在 sysHwinit 函数里会调用 sysSerialHwInit 来完成对各个串口的硬件初始化。在 usrRoot 中完成 SIO 设备的加载。

6) 网口

网口速度较快,一般单板都提供网口,可以用于调试,同时也可以用于单板与 PC 机之间的其他通讯,例如网管维护,程序/版本加载等。MPC850 的 SCC 口作为 10Mbps 以太网口时,UAS 以太网接口芯片采用 LXT908。在 VxWorks 中 SCC 作为以太网口的驱动程序在 motCpmEnd.c 中,一般不需要作修改,如果要修改 SCC 网口驱动,建议将 motCpmEnd.c 拷贝到 bspname 的目录下,然后修改 makefile。在网口发送中,必须先申请一个 mBlk-clBlk-cluster 的三元组,然后通过发送函数将数据包从 SCC 口发送出去。

```
END_CTRL* pDrvCtrl
M_BLK_ID pMblk
pMblk=netTupleGet(pDrvCtrl->endObject, pNetPool, END_
BUFSIZ, M_DONTWAIT, MT_DATA, FALSE)
motCpmEndSend(pDrvCtrl, pMblk)
```

4.3 MPC850 BSP 应用程序加载

一个可下载的应用程序包含一个或多个目标对象模块,这些模块可以下载和动态链接到 VxWorks,并从 shell 或调试器中启动。即允许目标模块装到一个运行的系统中。通过创建一个可下载的工程,构建该工程,使用适当的映像启动目标。下载工程所产生的部分链接和变换的.out 文件。一个可启动的应用程序包含一个链接到 VxWorks 映像的应用程序。目标机启动时,可启动的应用程序开始运行。通过创建一个可启动的工程添加应用程序文件,然后编辑 VxWorks 初始化文件 usrAppInit.c,为应用程序的初始化和启动工程添加调用。

4.3.1 构造一个基于 AMD79C97 网卡的 BSP

将下载得到的 AMD 公司的 vxworks\_end.exe 解压到 Torando 安装目录下的 Target 目录中进行覆盖,覆盖前请做好原 Target 的备份工作,便于恢复原开发环境,编译

新的网卡驱动:

定位到 tornado 目录执行 Tornado2.2\host\x86-win32\bin\torVars.bat;

定位到 tornado2.2\target\src\drv\end 目录执行 make CPU=PENTIUM tool=gnu ln97xend.o

定位到 tornado2.2\target\lib\pentium\PENTIUM\common 目录,并将上一步生成的文件 ln97xend.o 复制到此目录下。备份此目录下的文件 libdrv.a;

运行命令 arpentium-d libdrv.a ln97xEnd.o,删除 libdrv.a 中原有的 ln97xEnd 模块;

运行命令 arpentium-ra iOlicomEnd.o libdrv.a ln97xEnd.o 将刚刚创建的新 ln97xEnd 模块添加进去。

4.3.2 修改 Tornado 编译配置文件 config.c

定位到目录 C:\tornado2.2\target\config\pcPentium 并打开该目录下 Config.h 文件;

查找到定义 DEFAULT\_BOOT\_LINE 宏的地方,修改预处理条件 CPU==PENTIUM 分支下的

```
lhPci(0,0) your_host_name; d:\vxWorks h=192.168.80.169 e=192.168.80.254 u=target pw=target tn=target "
```

host:指定你的主机的名字,可任意指定;

d:\VxWorks:指定了 VxWorks 映象在主机上下载完整路径;

h=192.168.80.169:是宿主机的 IP 地址,可以根据实际情况修改;

e=192.168.80.254:是目标机的 IP 地址,可以根据实际情况修改;

u=target:指定了 FTP 服务器的用户名,这个 FTP 就是用来下载 VxWorks 映象的;

pw=vxworks:是 FTP 服务器与用户名 target 对应的密码;

tn=target:指定目标机的名字,任意指定即可。图 8 是 BootRom 中网卡的编译信息。

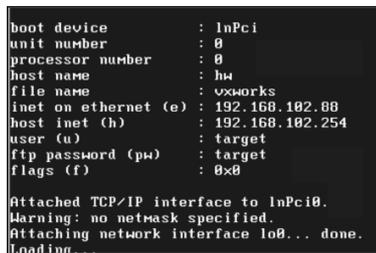


图 8 BootRom 中网卡的编译信息

4.3.3 编译 VxWorks 映像

编译生成 bootrom 后,还要创建一个 VxWorks 映像(image),也就是 VxWorks 操作系统本身的代码,如图 9,步骤如下。

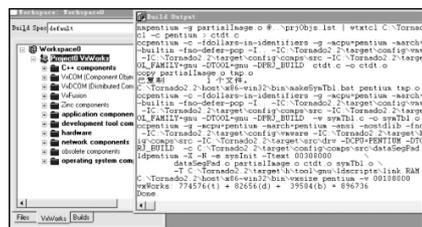


图 9 VxWorks 映像编译

创建一个“bootable VxWorks image”的工程,注意 BSP 选择 pentium;

打开工程,选择 VxWorks 组件,注意需要包括两个重要的组件:Telnet server 和 Target shell。前者使我们可以通过 Telnet 协议登录到 VxWorks 操作系统中;后者则可以让通过命令行控制 VxWorks 系统。另外,需要把所有 C++ 相关的选项都包含进去。

#### 4.3.4 建立调试环境

##### • 设置 FTP 服务器

主机上的 FTP 服务器用于在系统成功引导后,下载 VxWorks 的运行时映像,最好使用 Tornado 开发环境自带的 FTP 服务软件,如图 10 所示。FTP 需正确设置 Vx-works 映像路径和用户密码,与 config.h 中的 DEFAULT\_BOOT\_LINE 宏定义中的路径以及用户名密码保持一致。打开 FTP 的 LOG 功能,在 logging->Logging Option 中选择除 WinSock 以外的所有选项。正确设置完成后,可使用虚拟软盘或软盘映像启动 Vxworks 虚拟机,如图 11 所示。



图 10 FTP 上传 Vxworks 映像

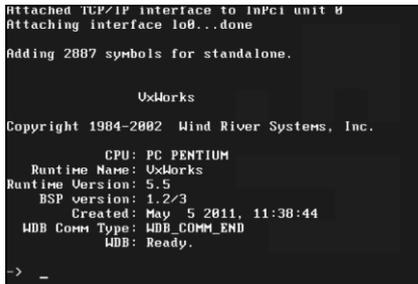


图 11 Vxworks 系统加载

由于在 VxWorks 映像中包含了 Telnet Server 的功能,因此在局域网范围内任何一台计算机都登录目标机查看目标机当前状态:在任何一台计算机命令行模式下执行 Telnet 192.168.102.254 命令。如图 12 显示 Telnet 目标板成功。

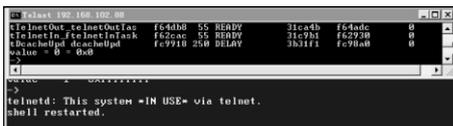


图 12 Telnet 目标板成功

##### • 配置 target server

打开 Tornado 开发环境,选择“Tools->Target Server->Configure”菜单,在“Description”中任意填写一个名字,这里是“Virtual Machine”;在“Available Back”中选择“wd-brpc”,并在下面的 IP 地址框中填写目标机的 IP 地址。配置完毕后,点击 Launch,启动 Target Server,从 Torando 下拉表中选中目标机,就可以看到各种标都己经激活,可以投入调试工作,如图 13 所示。可以进行的调试工作包括下载模块到目标机上运行,使用 Run Task 模式开启 Debugger

进行断点调试或单步跟踪,观察程序的动态运行情况等,如图 14 所示。

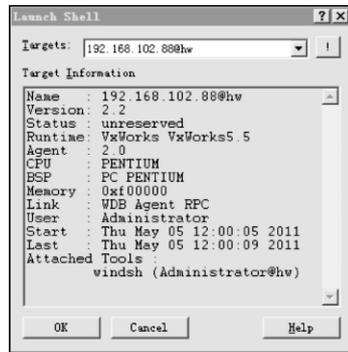


图 13 调试平台设置

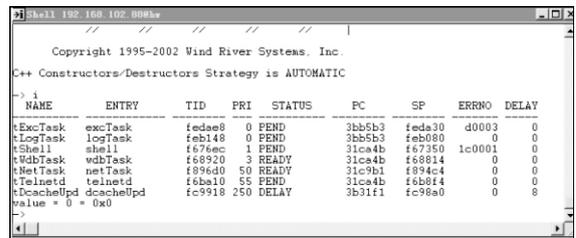


图 14 调试平台运行成功

## 5 结语

本文详细叙述了基于 MPC850 的 VxWorks 系统 BSP 调试平台的过程和使用,可以看出 VxWorks 工具的应用非常广泛,除了充当虚拟的目标机之外,还可以在多操作系统的开发、网络应用程序的开发、网络应用程序的测试、操作系统的功能测试等方面有很大的用途。在 VxWorks 开发中的上层应用开发,需要具备和真实设备基本类似的平台来进行调试、开发和测试工作,这对于 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] 徐惠民. 基于 VxWorks 的嵌入式系统及实验[M]. 北京:北京邮电大学出版社,2006.  
XU Huiming. Embeddedsystem experiment based on the Vx-Works[M]. Beijing: Beijing University of Posts and Telecommunications press,2006.
- [2] 刘志平. 基于 Vmware 虚拟网络的构建[J]. 内蒙古大学学报, 2007(1):94-98.  
LIU Zhiping. A Construction of Virtual Network Based on VMware Software[J]. Journal of Inner Mongolia University, 2007(1):94-98.
- [3] 杨少春. 采用 Vmware 构建虚拟并行计算网[J]. 计算机工程与设计,2006(14):2546-2547.  
YANG Shaochun. Constructing virtual parallel computing network with VMware[J]. Computer Engineering and Design, 2006(14):2546-2547.
- [4] 吕向阳,陈明义. 嵌入式系统创新实验室建设[J]. 实验室研究与探索,2005,24(5):32-33.  
LU Xiangyang, CHEN Mingyi. Building an Embedded System Innovative Laboratory[J]. Research and Exploration in Laboratory, 2005,24(5):32-33.

(下转第 75 页)

### 5.3 客户端的建立

客户端可以是 Web 页面也可以是一个独立的应用程序,由于 Web Service 是基于 SOAP(Simple Object Access Protocol)和 HTTP 这样的开放协议标准的,所以在客户端程序中使用 Web Service 提供的服务,就像调用本地的函数一样方便。用 C# 语言创建 Web Service 客户端代码如下: Client WEB page can also be a standalone application, Web Service is based on SOAP (Simple Object Access Protocol) and HTTP open protocol standard, so use the services provided by the Web Service client program, it is easy to like call the local function. C # language to create a Web Service client code as follows:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Windows.Forms;
namespace DataAccess
{
    public partial class MainForm : Form
    {
        public MainForm_Load()
        {
            AccessStruct();
        }
        private void AccessStruct()
        {
            localhost.TG_Resources_Server Database =
            new localhost.TG_Sources_Server();
            DataSet ds = Database.GetSources_2( "select
            * from 资源库 where 统计类别=金融 ");
            dataGrid1.DataSource = ds.Tables[0];
        }
    }
}
```

### 5.4 特点

统计信息提供者使用 Web Service 技术将统计资源封装成服务,并将其调用发布到互联网上,公布其调用接口,

统计信息需求者可以使用标准的 Web Service 调用方式来使用和再次封装这些资源,随着越来越多的资源提供者和需求者的加入,统计信息平台的规模会不断的扩大,资源的数量会不断的增多,同时由于公布了资源的调用接口,程序员可以利用这些接口开发出更多使用这些资源的客户端工具,使得整个系统规模会不断的扩大功能不断自我完善。Web Service 的相关标准都是 W3C 的开放协议,与平台和操作系统无关,不同的平台和操作系统上的 Web Service 的实现在很大程度上可以做到互操作,这就使异构平台上应用的集成变得很容易。

## 6 结语

随着网络技术的发展,软件产业的发展正从传统的提供软件产品的方式向提供软件服务的方向转变,而 Web 服务的出现正切合了这一理念的发展。把 Web 服务技术应用到统计信息共享平台的构建中,将系统中的各种资源和功能以 Web 服务的形式发布到互联网上,既有利于资源的共享,又能够使更多的用户参与到系统的不断完善中来。基于 Web Service 的统计信息平台的建立提高了系统的可扩展性和兼容性,增强了统计信息平台的适应能力,提高了系统的服务质量。

### 参考文献

- [1] Web Services Description Language (WSDL) Version 2.0. <http://www.w3.org/TR/wsdl20/>, 2007.
- [2] SOAP Version 1.2. <http://www.w3.org/TR/soap12-part1/>, 2003.
- [3] World Wide Web Consortium. <http://www.w3.org/2002/ws/>.
- [4] [www.microsoft.com/china/MSDN/li2brary/archives/library/dnppcgen/html/devtoolsMobileapps.asp](http://www.microsoft.com/china/MSDN/li2brary/archives/library/dnppcgen/html/devtoolsMobileapps.asp), 2003-081.
- [5] Newcomer E, Lomow G. Understanding SOA with Web Services[M]. Han Xu, translated. Beijing: Electronic.
- [6] LuoMin. Patterns: Service-Oriented Architecture and Web Services. <http://www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf>, 2004-4.
- [5] 黄晓玲, 段风云, 赵建科. 嵌入式系统实验教学体系的探索与实践[J]. 实验技术与管理, 2006, 23(4): 85-87.  
HUANG Xiaoling, DUAN Fengyun, ZHAO Jianke. Exploration and Practice of Experimental Teaching Architecture for Embedded System[J]. Experimental Technology and Management, 2006, 23(4): 85-87.
- [6] 王韬, 印勇, 刘国金. 基于 VxWorks 的嵌入式系统实验教学实验设计[J]. 实验室研究与探索, 2007, 26(1): 52-54.  
WANG Tao, YIN Yong LIU. Guojin. Embedded System Experimental Teaching Design Based on VxWorks[J]. Research and Exploration in Laboratory, 2007, 26(1): 52-54.
- [7] 冯先成, 李寒. 光纤到户远端设备的设计与测试[J]. 武汉工程大学学报, 2012, 34(4): 62-65.  
FENG Xiancheng, LI Han. Design and test of fiber to the home far-end equipment[J]. Journal of Wuhan Institute of Technology, 2012, 34(4): 62-65.
- [8] 冯先成, 李寒, 罗帆. 光纤到户 (FTTH) 的 ODN 工程设计及测试[J]. 计算机与数字工程, 2011(8): 171-173.  
FENG Xiancheng, LI Han, LUO Fan. Engineering Design and Test of FTTH ODN[J]. Computer & Digital Engineering, 2011(8): 171-173.
- [9] 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(1): 45-50.  
WANG Tao, YANG Shizhong, TAN Xiaoheng. Interrupt Process Design Based on MPC860 and VxWorks[J]. Telecommunication Engineering, 2005(1): 45-50.
- [10] 王柯. 基于 Pentium4 处理器的 VxWorks 板级支持包设计[J]. 电子科技, 2011(6): 57-59.  
WANG Ke. Design of BSP for Vxworks Based on Pentium4 CPU[J]. Electronic Science and Technology, 2011(6): 57-59.

(上接第 43 页)