基于 S698P4 的 VxWorks 操作系统定时器模块应用与开发

肖文斌
（珠海欧比特控制工程股份有限公司，广东 珠海 519080）

摘要：S698P4 是欧比特公司生产的基于 SPARC V8 架构的高性能的 32 位 RISC 嵌入式 4 核处理器。主要介绍基于 S698P4 硬件平台下 VxWorks 操作系统的定时器应用与开发。简要分析 VxWorks 的时钟和定时机制以及 S698P4 定时器模块的原理。将从 VIP 工程的建立到应用编程，讲述 VxWorks 定时器的应用与开发流程。从实验结果分析说明 S698P4 下 VxWorks 操作系统的定时器应用的正确性与可靠性，并得出相应结论。

关键词：S698P4；SPARC V8；VxWorks；定时器；VIP
中图分类号：TP303　　**文献标识码**：A　　**文章编号**：1000－8829(2014)02－0153－04

Application and Development of the Timer Module Under VxWorks Based on S698P4

XIAO Wen-bin
(Zhuhai Orbita Control Engineering Co., Ltd., Zhuhai 519080, China)

Abstract: S698PM is a high-performance 32-bit RISC embedded 4-core processor, which is based on SPARC V8 architecture, produced by Orbita. The application and development of the timer module are described under VxWorks based on S698PM hardware platform. The mechanisms of VxWorks clock and timing and the principle of S698P4 timer module are analyzed briefly. The application and development process of VxWorks timer are described from the establishment of VIP engineering to the application programming. The correctness and reliability of the timer application under VxWorks based on S698P4 are verified by experimental results, and the appropriate conclusions are drawn.

Key words: S698P4; SPARC V8; VxWorks; timer; VIP

VxWorks 操作系统是美国 Wind River 公司于 1983 年设计开发的一种嵌入式实时操作系统（RTOS），高性能的内核以及友好的用户开发环境，使其广泛应用在通信、国防、工业控制、医疗设备等嵌入式领域，特别是在现代各种嵌入式计算机系统中，比如军事指挥系统、武器控制系统或工业控制系统中。在这些系统中，往往不是单个计算机的控制，而是多个计算机以及其他设备组成通信网络共同完成控制作用，因此，网络对于嵌入式操作系统极为重要。

S698P4 是基于 SPARC V8 架构的高性能的 32 位 RISC 嵌入式 4 核处理器。采用 SMP “对称多处理”技术，是在一个内核里集成 4 个功能一样的处理器核心，各 CPU 之间共享内存子系统以及总线结构，总线竞争和仲裁由硬件自动完成，不需要用户设置。它专为嵌入式应用而设计，具有高性能、低复杂度和低功耗的特点。

本文主要讲述 VxWorks 的时钟和定时机制、S698P4 定时器和 S698P4 驱动应用编程及运行结果。

1　VxWorks 的时钟和定时机制

在 VxWorks 操作系统中，进行规定时间的操作是很常见的。比如周期性地执行某个任务或者在某个时间点进行某项规定的操作，这需要有非常精确的定时器作为时间基准，在 VxWorks 内部，任务之间的调度运行激励也需要有非常精确的定时，内核周期性地执行时钟中断。在 VxWorks 操作系统中，提供两种基于

收稿日期：2013－11－18
作者简介：肖文斌（1987—），男，广东韶关人，本科，工程师，主要研究方向为 SPARC V8 架构处理器嵌入式软件开发。

硬件的定时时钟：系统时钟（system clock）和系统辅助时钟（system auxiliary clock）。当然，VxWorks 中也支持 ANSIC C 和 POSIX 标准接口的时钟机制。

（1）系统时钟。

主要的相关函数有：

① sysClkConnect()，绑定时钟中断程序。

```
STATUS sysClkConnect
(
  FUNCPTR routine,   /* 要绑定的时钟处理程序 */
  int arg   /* 处理程序的参数 */
)
```

② sysClkDisable()，关闭系统中断。

```
void sysClkDisable(void)
```

③ sysClkEnable()，打开系统中断。

```
void sysClkEnable (void)
```

④ sysClkRateGet()，获得系统时钟频率。

```
void sysClkRateGet(void)
```

⑤ sysClkRateSet()，设置系统时钟频率。

```
STATUS sysClkRateSet
(
int ticksPerSecond /* 每秒的时钟中断数或者 tick 数 */
)
```

（2）系统辅助时钟。

辅助时钟就是对于系统时钟的辅助和补充，它在实时系统中也经常被使用。当需要较高的时钟频率时，由于系统时钟还有很多其他的用途，如为系统调度提供基准等，因此其频率不能够设置得太高，这时可以使用辅助时钟。辅助时钟也是基于硬件定时器的，所以比较精准。辅助时钟如果可用则系统时钟也必须可用。辅助时钟在其他一些参考书上被称为主时钟的“影子”，形象地说明了它的作用。类似的，跟辅助时钟相关的函数主要有：

① sysAuxClkConnect()，绑定辅助时钟中断处理程序。

```
STATUS sysAuxClkConnect
(
  FUNCPTR routine,   /* 要绑定的时钟处理程序 */
  int arg   /* 处理程序的参数 */
)
```

② sysAuxClkDisable()，关闭辅助时钟中断。

```
void sysAuxClkDisable(void)
```

③ sysAuxClkEnable()，打开辅助时钟中断。

```
void sysAuxClkEnable(void)
```

④ sysAuxClkRateGet()，获取辅助时钟中断频率。

```
void sysAuxClkRateGet(void)
```

⑤ sysAuxClkRateSet(void)。

```
STATUS sysAuxClkRateSet
(
  int ticksPerSecond   /* 每秒的时钟中断数或者 tick 数 */
)
```

从上面的辅助时钟系统函数可以看出，它们和系统时钟函数非常相似，是对系统时钟很好的辅助和补充。

2　S698P4 定时器

S698P4 系统平台提供了 2 个 32 位定时器，提供硬件定时中断，分别为定时器 1、定时器 2。其中，定时器 2 可作为看门狗使用。

通用定时器应用在一个预置数寄存器和 1 ~ 7 个递减计时器。通用定时器作为 AMBA APB 总线从设备。定时器能够在定时器下溢出时出现中断。

预置数寄存器被系统时钟控制而且在每个时钟周期上递减。当预置数寄存器下溢出，它会从预置数重载寄存器重新装载，同时产生一个定时器滴答（tick）。定时器共享递减器以保存区域。在下一个定时器滴答时，下个定时器会以等价（预置数重载寄存器值 + 1）的有效衰减速率逐渐减少。

控制寄存器控制定时器的操作。一个定时器通过设置控制寄存器中使能位使能操作。然后，定时器计数器的值在每个预置数滴答后渐减。当一个定时器计数器下溢出，如果控制器中的复位位为 1 时，它将会自动地重载对应的定时器重载寄存器中的值；否则，它将会停止在 -1 而且复位使能位。

定时器作为一个中断。当具有中断使能的任一个定时器计数器下溢出时，定时器单元将会产生共享中断。如果配置成每个定时器发出各自的中断信号，定时器单元将会在一个定时器单元下溢出时，向适当的线上发送中断信号（如果给现在的定时器的中断使能位被置位）。在下溢出定时器的控制寄存器的中断未决位将置位，并保持，直到通过写‘0’清除。

定时器分享相同的递减器。通过设定控制寄存器的链接位，定时器 n 能够与前一个定时器 $n-1$ 配合使用。当定时器 $n-1$ 下溢出时，定时器 n 计数器值减 1。

当向控制寄存器的载入位写‘1’时，每个定时器都会重载重载寄存器中的值。定时器 2 可以当作看门狗使用，当定时器 2 计数器下溢出时，就驱动一个看门狗输出信号。

注意：当定时器 2 作看门狗使用时，不能把定时器 2 作其他用途。

寄存器描述如表 1 所示。

3　基于 S698P4 处理器的 VxWorks 定时器应用开发

基于 S698P4 处理器的 VxWorks 定时器应用开发

是在 Workbench 环境下进行的。VxWorks 操作系统版本为 6.7。开发流程依次是创建 VIP(VxWorks image project)工程、应用编程和执行结果分析。

表 1 通用定时器寄存器

地址	预置数
0x80000300	预置数
0x80000304	预置数重载值
0x80000308	配置寄存器
0x8000030C	未使用
0x80000310	定时器 1 计数寄存器
0x80000314	定时器 1 重载值寄存器
0x80000318	定时器 1 控制寄存器
0x8000031C	未使用
0x80000320	定时器 2 计数寄存器
0x80000324	定时器 2 重载值寄存器
0x80000328	定时器 2 控制寄存器

(1) 创建 VIP 工程。

在 Workbench 中点击“File -> New -> VxWorks Image Project”新建 VIP 工程,工程名称为 S698P4_Timer。再根据 S698P4 开发板硬件信息选择“s698p4”BSP 包和“gnuv8”编译工具,完成工程创建。

(2) 应用编程。

完成 VIP 工程创建后,需要对工程进行相关配置,配置 VxWorks 的 kernel。依据 S698P4 开发板上的硬件信息修改以下参数:在“Component Configuration”窗口中选择“hardware (default)” -> “memory (default)” -> “BSP Memory Configuration”,修改其中的“RAM high Address”为 0x40403000,“RAM low Address”为 0x40003000,“local memory address”为 0x40000000,将“LOCAL_MEM_SIZE”改为 0x003ff000;再到“Project Explorer”窗口中选择 S698PM_Timer 工程下的“s698p4”,双击打开目录下的 config.h 文件,修改“RAM_HIGH_ADRS”为 0x40403000,“RAM_LOW_ADRS”为 0x40003000,“LOCAL_MEM_LOCAL_ADRS”为 0x40000000,“LOCAL_MEM_SIZE”为 0x003ff000。如图 1 所示。

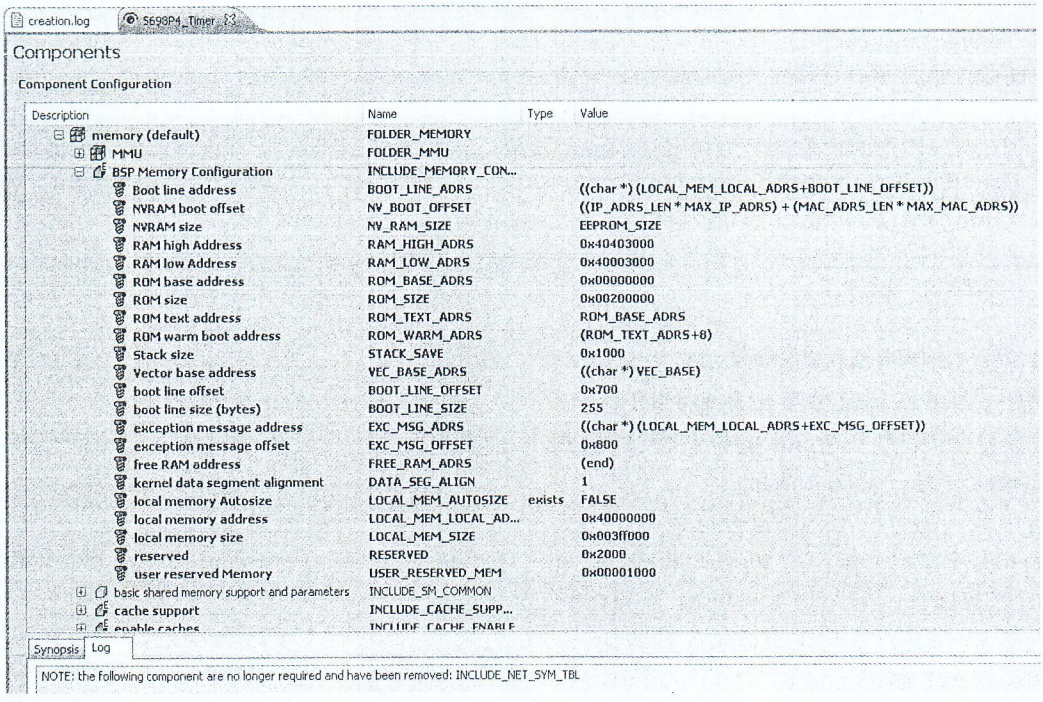

图 1 修改 BSP Memory Configuration

在 VxWorks 中,S698P4 的第 1 个定时器用于系统时钟(sysClk),第 2 个定时器用于辅助时钟(auxClk)。而 VxWorks 中的时间戳(stamp clock)将与系统时钟进行竞争共用。

利用 VxWorks 中的相关定时器的 API 读取系统时钟、辅助时钟和时间戳的相关信息,分别打印出来,应用流程图如图 2 所示。利用辅助时钟产生设定时间内的中断,用于精确执行任务。流程图如图 3 所示。

4 S698P4 定时器驱动应用执行结果

将 VIP 工程编译生成 VxWorks 镜像文件后,通过 cygwin 下载到 S698P4 开发板中运行。下载前,将 S698P4 的串口 1(UART0)与 PC 机的串口连接,并打开串口调试助手,将串口调试助手的波特率设置为 38400。下载命令为

```
v8mon.exe -i -u -eth -freq 20
```

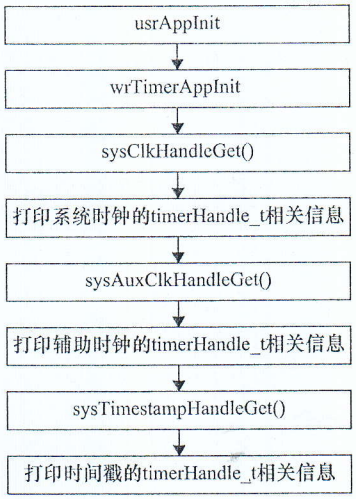

图 2　获取时钟相关信息流程图

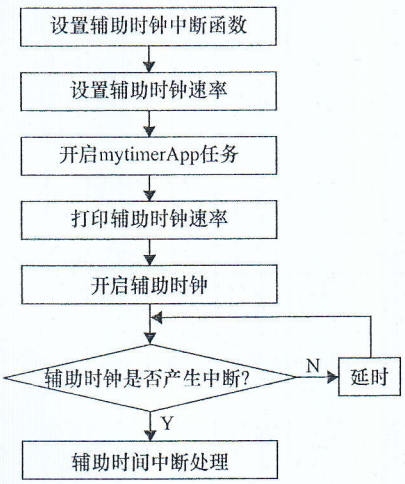

图 3　辅助时钟应用流程图

下载完成后，在串口调试助手中将会打印出 VxWorks 的相关信息，并作为 VxWorks 的控制台，通过此控制台与 VxWorks 交互。如图 4 所示。

在控制台中输入命令【d 0x80000300】便可以显示内存 0x80000300 上的内容。这里 0x80000300 为 S698P4 的定时器基地址。对照表 1 S698P4 通用定时器寄存器信息便可以知道此时，定时器 1 已经使能，定时器 2 已关闭。如图 5 所示。

5　结束语

S698P4 是欧比特公司针对实时应用的嵌入式领域研制的基于 SPARC V8 架构的 4 核高性能、低功耗 32 位处理器。S698P4 提供了内部看门狗、定时器、中断控制器以及串行通信接口；针对航空航天领域，S698P4 提供了 CAN 总线接口及以太网接口。详细介绍了在 VxWorks 下 S698P4 定时器的应用开发。从 VxWorks 的 VIP 工程建立，到 POSIX Timer 应用分析，再到运行结果分析。详细分析了 S698P4 在 VxWorks 中的应用。

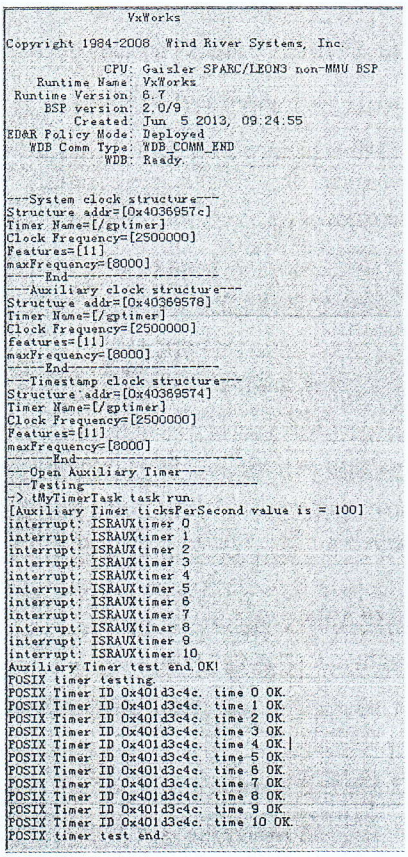

图 4　Timer 数据结构

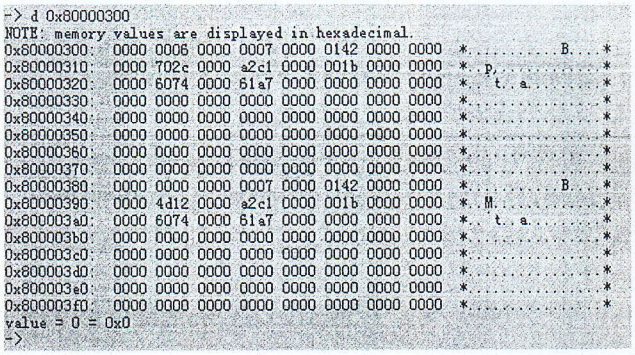

图 5　定时器寄存器

参考文献：

[1] 珠海欧比特控制工程股份有限公司. S698P4 四核并行处理器芯片用户手册[Z]. 2011.

[2] 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.

[3] Aeroflex Gaisler AB. VxWorks-drivers-6. 7-1. 1. 2. 0[Z]. 2009.

[4] Wind River. VxWorks BSP Developer's Guide, 6. 7[Z]. 2008.

[5] Wind River. VxWorks Application Programmer's Guide, 6. 7[Z]. 2008.

[6] Wind River. VxWorks Kernel Programmer's Guide, 6. 7[Z].

□