基于 S698PM 的 VxWorks 操作系统串口模块应用与开发

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：S698PM 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。介绍了基于 S698PM 硬件平台下 VxWorks 操作系统的串口模块应用与开发，简要分析了 VxWorks 的 I/O 系统及其相关操作，并从 VIP 工程的建立到应用编程，讲述 VxWorks I/O 系统中的串口模块的应用与开发流程。实现 S698PM 平台下 VxWorks 操作系统的串口通信，并得出相应结论。

关键词：S698PM；SPARC V8；VxWorks；I/O 系统；VIP
中图分类号：TP303　　**文献标识码**：A　　**文章编号**：1000－8829(2014)01－0126－03

Application and Development of the UART Module Under VxWorks Based on S698PM

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 UART module under VxWorks based on S698PM hardware platform are introduced. The I/O system and related operation of VxWorks are analyzed briefly. The application and development process of VxWorks I/O system serial modules are described from the establishment of VIP engineering to the application programming. The serial communication of VxWorks based on S698PM platform is realized, and the appropriate conclusion is drawn.

Key words: S698PM; SPARC V8; VxWorks; I/O system; 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，使其已广泛应用在通信、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。

S698PM 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，带 MMU，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。

这里分 3 个步骤来讲述基于 S698PM 处理器的 VxWorks 操作系统定时器模块应用与开发：① 创建 VIP（VxWorks image project）工程、RTP（VxWorks real time process project）工程和 ROMFS（VxWorks ROMFS file system project）工程；② S698PM 驱动应用编程；③ S698PM 驱动应用程序运行结果。

1　VxWorks 的 I/O 系统

在 VxWorks 中，I/O 系统向用户屏蔽了硬件层，为用户提供了一个统一的标准接口。对于应用层开发者来说，只要了解了 I/O 系统标准接口的使用方法，就可以正确地操作外部设备。但对于底层驱动工程师而言，必须要全面地了解 I/O 系统，才能够更好地为上层应用开发提供良好的支持。

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。

如果从数据的流向来讲，I/O 系统可以分为 3 个层次：面向用户的 I/O 接口层、硬件驱动层和具体硬件。这 3 个层次一起构成了 I/O 系统。

VxWorks 支持的 I/O 设备基本上可以分为字符设备和块设备。字符设备主要指以字符为单位进行数据操作的设备，如串口等；块设备是以大批量的“块”为单位进行数据操作的设备，如网络设备、存储设备等。字符设备直接通过 I/O 接口和应用层进行交互，而块设备要通过文件系统与 I/O 接口层进行交互。

串口设备属于 I/O 设备中的字符设备，通过基本的 I/O 接口便可以对其进行操作。在 VxWorks 中，设备都是按照文件来操作的，打开一个设备需要知道它的文件名，当成功打开某个设备时，会返回一个文件描述符。

基本的 I/O 函数介绍如下。

(1) creat()：创建一个文件。

```
int creat
(
  const char * name,  /* 创建的文件名 */
  int flag  /* O_RDONLY,O_WRONLY,OR O_RDWR */
)
```

creat()函数在指定的设备上创建一个文件并且打开这个文件，创建成功后则返回文件描述符。其中 flag 选项可以为 O_RDONLY(只读操作)、O_WRONLY(只写操作)或 O_RDWR(可读可写操作)。

(2) open()：打开一个文件。

```
int open
(
  const char * name,  /* 要打开的文件名 */
  int flag,  /* O_RDONLY,O_WRONLY,O_RDWR or O_CREAT */
  int mode  /* 文件模式 */
)
```

通常，用 open()函数打开的是已经存在的文件，对于 NFS 文件，需要设置 mode 参数。其中 flag 选项可以为 O_RDONLY(只读操作)、O_WRONLY(只写操作)、O_RDWR(可读可写操作)或 O_CREAT(创建文件)。

(3) remove()：删除一个文件。

```
STATUS remove
(const char * name  /* 要删除的文件名 */)
```

(4) close()：关闭一个文件。

```
STATUS close
(int fd  /* 要关闭的文件描述符 */)
```

如果操作成功后返回 OK。

(5) read()：从文件或设备上读取数据。

```
int read
(
  int fd,  /* 要读的文件描述符 */
  char * buffer,  /* 接收数据的 buffer 指针 */
  size_t maxbytes  /* 读取到缓冲中的最大字节数 */
)
```

如果文件描述符不存在、没有读操作的驱动函数、设备读操作返回错误都会返回 ERROR，否则返回所读的字节数。

(6) write()：向一个文件写数据。

```
int write
(
  int fd,  /* 要写的文件描述符 */
  char * buffer,  /* 要输出的数据缓冲指针 */
  size_t nbytes  /* 要写的字符数 */
)
```

如果文件描述符不存在、没有写操作的驱动函数、设备写操作返回错误都会返回 ERROR，否则返回所写入的字节数。

(7) ioctl()：执行 I/O 控制函数。

```
int ioctl
(
  int fd,  /* 文件描述符 */
  int function,  /* 函数代码 */
  int arg  /* 传入的参数 */
)
```

2 S698PM 串口应用开发

基于 S698PM 处理器的 VxWorks 串口应用开发是在 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 工程、应用编程和执行结果分析。

2.1 创建 VIP 工程

在 Workbench 中单击“File”→“New”→“VxWorks Image Project”新建 VIP 工程，工程名称为 S698PM_UART。再根据 S698PM 开发板硬件信息选择“s698pm”BSP 包和“gnuv8”编译工具，完成工程创建。

2.2 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 S698PM V6 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 0x62003000“RAM low Address”为 0x60003000，“local memory address”为 0x60000000。将“VX_SMP_NUM_CPUS”改为 4。再到“Project Explorer”窗口中选择工程下的“s698pm”，双击打开目录下的 config.h 文件，修改“RAM_HIGH_ADRS”为 0x62003000，“RAM_LOW_ADRS”为

0x60003000，“LOCAL _ MEM _ LOCAL _ ADRS”为 0x60000000。

然后对 S698PM 的 UART1 进行设置，将其波特率设置为 38400 bit/s，串口工作模式为 RAW_MODE，UART 应用测试流程如图 1 所示。

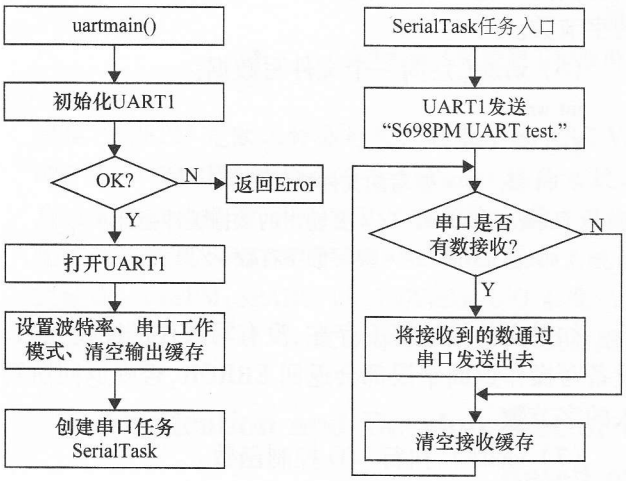

图 1　UART 应用流程图

在 VxWorks 中对 UART 主要有以下操作程序：

① 打开串口 1。

```
sd = open("/tyCo/1",O_RDWR,0);
```

② 设置串口波特率、工作模式和清除输入输出缓冲。

```
status = ioctl(sd,FIOBAUDRATE,38400);//设置串口波特率
status = ioctl(sd,FIOSETOPTIONS,OPT_RAW);//设置串口工作模式
status = ioctl(sd,FIOFLUSH,0);//清空输入输出缓冲
```

③ 串口发送。

```
write(sd,send_buffer,sizeof(send_buffer));
```

④ 串口接收。

```
read(sd,receive_buf,1);
```

2.3　执行结果分析

编译 S698PM_UART 工程，然后通过 cygwin 下载到目标板上运行。通过串口助手打印 VxWorks 控制台，在控制台中可以运行【rtp】命令查看正在运行的 RTP 任务。图 2 为串口调试助手打印的 VxWorks 控制截图。

此时对 S698PM 的 UART1 进行串口发送与接收测试，测试结果如图 3 所示。

3　结束语

S698PM 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 架构的高性能、低功耗的 32 位4核处理器。S698PM提供了内部看门狗、定时器、

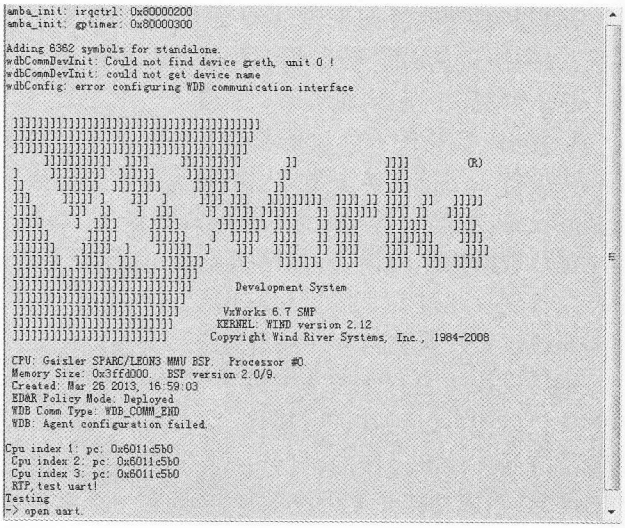

图 2　运行控制台截图

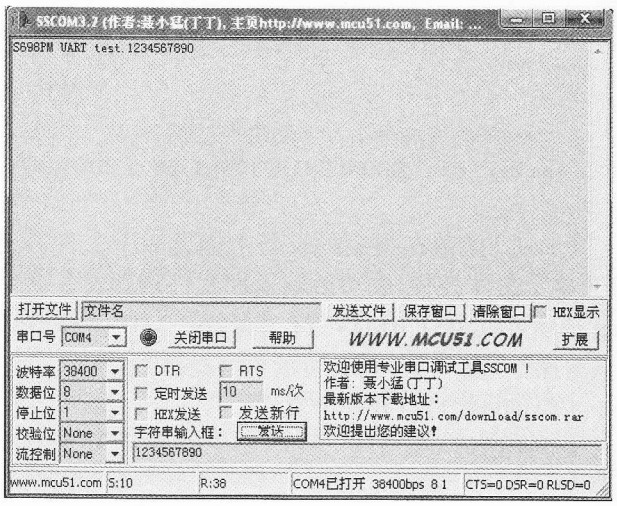

图 3　运行串口发送与接收截图

中断控制器以及串行通信接口；针对航空航天领域，S698PM 提供了 CAN 总线接口及以太网接口。详细介绍了在 VxWorks 下 S698PM UART 的应用开发。从 VxWorks 的 VIP 工程建立，到 UART 应用例程分析，再到运行结果分析。详细分析了 S698PM 在 VxWorks 中的应用。

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.

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

[3]　Wind River Systems, Inc. VxWorks BSP Developer's Guide, 6.7[Z]. 2008.

[4]　Wind River Systems, Inc. VxWorks Application Programmer's Guide, 6.7[Z]. 2008.

[5]　Wind River Systems, Inc. VxWorks Kernel Programmer's Guide, 6.7[Z]. 2008.

□