

文章编号:1009-671X(2004)11-0021-02

Vxworks 中串行通信的研究

张伟伟,文爱军,钟 宁

(西安电子科技大学 综合业务网国家重点实验室,陕西 西安 710071)

摘 要:介绍了嵌入式操作系统 Vxworks 的特点及串行通信的概念,从 Vxworks 的输入/输出系统的特点着手,讨论了 Vxworks 下串行设备的通信机制,解决了在 Vxworks 下实现串行通信的问题,并且给出了程序示例.

关 键 词:嵌入式操作系统;Vxworks;串行通信

中图分类号:TP311 **文献标识码:**A

Reserach on serial communication in Vxworks

ZHANG Wei-wei, WEN Ai-jun, ZHONG Ning

Abstract: The characteristics of embedded operating system Vxworks and the concept of serial communication are briefly introduced. The mechanism of the serial communication is discussed. Finally, the problem of serial commucation in Vxworks is resolved and example routines are given.

Key words: embeded operating system; Vxworks ; serial communication

随着信息技术和互联网的飞速发展,以及计算机、通讯、数码产品等领域的高速增长,数字化时代已经来临. 嵌入式设备是数字化时代的主流产品,嵌入式软件是数字化产品的核心,作为嵌入式软件的基础和关键,嵌入式操作系统在产业发展过程中扮演着越来越重要的角色. 应用遍及工业自动化、网络通信、航空航天、医疗仪器等领域.

1 Vxworks 的特点

Vxworks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统 (RTOS), 是一个运行在目标机上的高性能、可裁减的嵌入式实时操作系统. 它有以下几个特点: 高性能的微内核设计、兼容 POSIX 1003.1b 标准、综合的网络工具、支持多种开发和运行环境、提供多种操作系统选项.

2 Vxworks 中的串行通信

2.1 串行通信及串口

串行通信是指外设和计算机间使用一根数据信号线(另外需地线,可能还需要控制线),数据

在一根数据信号线上一位一位地进行传输,每一位数据都占据一个固定的时间长度. 这种通信方式使用的数据线少,在远距离通信中可以节约通信成本,但其传输速度比并行传输慢.

串口是计算机一种常用的接口,连接线少,通讯简单,得到广泛的使用. 常用的串口是 RS-232-C 接口(又称 EIA RS-232-C),它是在 1970 年由美国电子工业协会(EIA)联合贝尔系统、调制解调器厂家及计算机终端生产厂家共同制定的用于串行通讯的标准. 全名是“数据终端设备(DTE)和数据通讯设备(DCE)之间串行二进制数据交换接口技术标准”,该标准规定采用一个 25 针的 DB25 连接器,对连接器的每个引脚的信号内容加以规定,还对各种信号的电平加以规定.

2.2 Vxworks 输入/输出系统

Vxworks 操作系统的 I/O 系统可以提供简单、统一、与任何设备无关的接口. 这些设备包括: 面向字符设备、随机块存储设备、虚拟设备、控制和监视设备以及网络设备. Vxworks 的 I/O 系统包括基本 I/O 系统和缓冲 I/O 系统. Vxworks 的 I/O 系统具有比其他 I/O 系统更快速,兼容性更

好的特性. 这对于实时系统是很重要的.

在 Vxworks 中, 应用程序通过打开指定的文件来操作 I/O 设备. 一个指定的文件可以是一个非结构化的原始设备, 如: 一个串行通信通道或一个任务间管道. 也可以是随机存储设备上的一个逻辑文件. 例如: /usr/myfile 表示在一个名为 /usr 上的一个叫 myfile 的文件(注: 以斜线 '/' 开始的名称表示基于 NFS 文件系统的网络设备, 如: /usr); /pipe/mypipe 表示一个名为 mypipe 的管道; /tyCo/0 表示串行通道(串口 1).

由于在 VxWorks 中, 将 I/O 系统设计成为与设备类型无关的统一接口, 因此对于串行口的操作仍然可以视为对一个文件的操作, 而不必关心设备或程序驱动实现的细节. 因此在 Vxworks 中实现串行通信非常容易.

2.3 串口通信程序示例

在串口通信程序的设计中, 首先安装串行驱动程序, 并创建串口设备, 然后打开串口并对其通信参数进行配置, 接着就可以依据串口打开时的读写标志, 调用函数 write()、read() 对串口进行读写操作了. 对于串口通信, 仍然要关心数据接收的实时性. 可采用中断方式, 利用 VxWorks 提供的 select 函数的事件触发机制, 将读串口的任务阻塞使其一直等待数据, 当有数据来到的时候该任务会立刻自动响应, 提高系统的实时性.

1) 初始化并且创建一个串口设备

```
ttyDrv(); /* 初始化串口设备 */
pSioChan = sysSerialChanGet(0); /* 创建串口设备 */
ttyDevCreate(/tyCo/0, pSioChan, 512, 512);
```

2) 打开串口, 并设置串口参数:

a) 打开串口是通过标准的文件标识符来进行操作的

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

b) 设置串口参数

通过调用 ioctl() 函数执行 SIO_HW_OPTS_SET 功能可以改变设备驱动程序的硬件配置(改变停止位位数, 奇偶校验方式). 在文件 "install-Dir/target/h/sioLib.h" 中定义了设备配置所需的 POSIX 标准常量: CS8, STOPB, PARENB, PARODD 等.

· 设置串口数据模式

```
ioctl ( SerialDevFd, FIOSETOPTIONS, OPT _
```

```
RAW&OPT_RAW);
```

· 设置波特率

```
ioctl ( SerialDevFd, FIOBAUDRATE, 4800);
```

· 设置数据位数

```
ioctl ( SerialDevFd, SIO _ HW _ OPTS _ SET, CS8);
```

· 设置校验位

```
ioctl ( SerialDevFd, SIO _ HW _ OPTS _ SET, PARENB); //使能校验
```

```
ioctl ( SerialDevFd, SIO _ HW _ OPTS _ SET, PARODD); //设置校验方式
```

· 设置停止位

```
ioctl ( SerialDevFd, SIO _ HW _ OPTS _ SET, STOPB); //2 位停止位(不指明是 1 位)
```

注: 其实串口配置一般可以使用默认值(BBBB 为波特率: 9600, P 为奇偶检验: n, D 为数据位: 8, S 为终止位: 1), 只要修改波特率就行了.

3) 写串口

```
write ( SerialDevFd, SendBuf, DataLen);
```

4) 读串口

利用 select 函数的事件触发机制, 将读串口的任务阻塞, 使其一直等待数据, 当有数据来到的时候该任务会立刻自动响应. 它不仅支持任务同时等待对多个设备进行 I/O 操作, 还能够指定等待过程的最长时间. 通过 select() 函数可以指定某一个任务等待哪些文件描述符变为活动状态. 在 select() 函数中使用位域标志设置有关文件的读/写状态. 通过 select() 函数的返回值反映相应的文件描述符是否有效. 因此任务能够同时监视多个串口, 一旦其中一个串口来了数据, 任务就被激活了, 并从这个串口读取数据. 这里以一个任务同时监视两个串口为例, 读取串口上的数据, 假定 COM1 文件描述符为 fds[0], COM2 的文件描述符为 fds[1], 这里只是列出了读操作过程的程序.

```
FOREVER
```

```
{
```

```
    FD_ZERO (&readFds); /* 设备集合中的所有读标志位清 0 */
```

```
    FD_SET (fds[0], &readFds); /* 初始化控制位 */
```

```
    FD_SET (fds[1], &readFds);
```

(下转第 44 页)

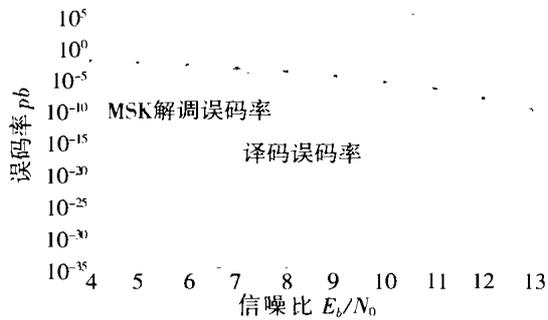


图 3 误码率曲线

10^{-10} 、达到了令人满意的效果;而在信噪比小于 7 dB 以下,编码增益变为负值,性能有很大的恶化,这说明了解调输出的误码率已经超过了(31, 15) RS 码的纠错能力,译码增加的冗余符号只消耗了能量,并没有带来好处。所以在实际工程中应用纠错码时必须对信噪比提出一个最低要求,若低于此值,则必须设法改善信道质量,以达到要求,否则就不能应用纠错编码来改善信道质量。

4 结束语

通信技术的飞速发展,对信息的传输、处理性能的要求也越来越高。正是由于 RS 编码技术能特别有效地纠正突发错误,而且纠错码随着分组大小的增加而纠错性能更加有效,其实现的复杂度要低于相同长度的其他编码方法,也正是基于这些优点,RS 码具有很大的吸引力,因此在未来的通信、计算机系统等各个领域中的应用范围会更加广泛。

参考文献:

- [1] 王新梅. 纠错码—原理与方法[M]. 西安: 西安电子科技大学出版社, 2002.
- [2] 小克拉克 G C, 凯恩 J B, 著. 数字通信系统中的纠错编码[M]. 聂涛, 译. 北京: 国防工业出版社, 1988.
- [3] 刘徐德. 战术通信、导航定位和识别综合系统文集: 1 [C]. 北京: 电子工业出版社, 1991.

[责任编辑:李雪莲]

(上接第 22 页)

```
width = ((fds[0] > fds[1]) ? fds[0] : fds[1]);
if (select (width + 1, &readFds, NULL, NULL, NULL) == ERROR)
    /* 程序处于阻塞状态,等待其中的一个串口变为可用状态 */
    return (ERROR);
for (i=0; i < 2; i++) /* 从可用的串口读取数据 */
    if (FD_ISSET (fds[i], &readFds))
        ReadLen = read (fds[i], buffer, MAX_DATA);
        printf("Reading Data from COM%d: %s\n", i + 1, buffer);
        while(ReadLen != 0) /* 继续读取数据 */
            ReadLen = read (fds[i], buffer, MAX_DATA);
            if(ReadLen != 0)
                printf("Continue reading Data from
```

```
COM%d: %s\n", i + 1, buffer);
```

3 结束语

Vxworks 是一种高性能嵌入式实时操作系统,它以其良好的可靠性和卓越的实时性被广泛地应用在通信、自动化、航空等各个领域。由于在 VxWorks 中,将 I/O 系统设计成为与设备类型无关的统一的接口,因此对串行的操作可以视为对一个文件的操作,而不必关心设备或程序驱动实现的细节,实现起来非常方便。许多外部终端设备如打印机、逻辑分析仪等都采用串行方式,以及对板子的调试也要用到串行口,因此掌握在 Vxworks 下的串行通信还是很有必要的。

参考文献:

- [1] PETER W. Gofton 精通串行通信[M]. 王仲文,译. 北京:电子工业出版社,1995.
- [2] TOMADO,孔祥营,柏桂枝. 嵌入式实时操作系统 Vx-Works 及其开发环境[M]. 北京:中国电力出版,2002.

[责任编辑:李雪莲]