第33卷 第17期 计 算 机 工 程 2007年9月
Vol.33 No.17 Computer Engineering September 2007
·开发研究与设计技术· 文章编号：1000—3428(2007)17—0275—03 文献标识码：A 中图分类号：TP393.04

基于MPC8270的VxWorks下串口驱动程序开发

吴雨舟，路唯佳，张 平
(北京邮电大学无线新技术研究所，北京 100876)

摘 要： MPC8270是目前网络和通信领域应用非常广泛的一款通信处理器，该文阐述了利用MPC8270的SMC(串行管理控制器)在UART传输协议下进行串口通信的实现方法。给出了串口驱动程序的结构以及在VxWorks的BSP(板级支持包)中如何对串口驱动程序进行加载的原理。该串口驱动经过长时间的应用和测试，在整个系统中运行稳定。

关键词： 串口驱动程序；UART；VxWorks；SMC

Development of Serial Ports Driver Based on MPC8270 in VxWorks

WU Yu-zhou, LU Wei-jia, ZHANG Ping
(Wireless Technology Innovation Institute, Beijing Univercity of Posts and Telecommunications, Beijing 100876)

【Abstract】 The MPC8270 is a versatile communications processor that is widely used both in network and communication system. This paper expounds how to communicate between UART serial ports by using the SMC (serial management controller) of MPC8270. It also analyzes the structure of the serial port driver and how to load the serial port driver in the VxWorks BSP (board support package). This serial ports driver which has been applied and tested for a long time runs steadily in the whole system.

【Key words】 serial ports driver; UART; VxWorks; SMC

本文的开发平台PMC(PCI mezzanine card)8270板是为国家“十五”“863”“Beyond 3G蜂窝移动通信无线网络试验系统研究开发”项目演示系统所设计的CPU子板。PMC8270板是基于PowerQuicc II通信处理器的嵌入式硬件平台。CPU的60x总线通过PMC接口连接基于Xilinx Vertex II pro FPGA的ATCA处理母板。板载1MB+16MB FLASH以及32MB SDRAM，提供可靠的UART串口及FE快速以太网的支持，并提供一个JTAG口供CPLD程序下载及CPU硬件调试。

串口是PMC8270板的调试端口和低速数据通信接口。由于PMC8270板没有显示终端和输入设备，因此在调试过程中需要将PMC8270板的信息通过串口重定向到显示终端。此外，在系统运行过程中，还可以通过串口配置一些受控设备，通过串口接收监控设备所传来的测量数据，并通过串口向受监控设备发送命令字。

1 MPC8270处理器

PMC8270板以Motorola PowerQuicc II系列MPC8270为中央处理器，主要由3个部分组成：嵌入式PowerPC内核(G2_LE核)，系统接口单元(SIU)，通信处理模块(CPM)。

CPM包括一个32位RISC结构的通信处理器CP，主要完成底层的通信任务处理，使得PowerPC核可以更高效地处理上层的任务，同时还包括3个快速串行通信控制器FCC、2个多通道控制器MCC、4个串行通信控制器SCC、2个串行管理控制器SMC、1个串行外围接口SPI和1个I^2C总线控制器。支持的通信协议包括ATM、HDLC、Fast Ethernet以及UART等，可以根据需要选择相应的控制模块进行配置，完成不同的功能。此外，这种双处理器结构功耗也要低于传统结构的处理器。

在本驱动程序中将串口所采用的通信协议设计为UART异步通信协议。在设计时需要注意的一点是在SCC下的UART协议在功能和性能上都优于SMC下的UART协议，因此，SMC设备更多地应用在对目标板的调试和监控中，这样就可以将SCC用来完成其他的更复杂的功能。在本驱动程序中将CPM的SMC1配置成UART控制器来实现设备间通过串口的通信。

2 SMC中的UART协议简介

UART协议是典型的面向字符的协议，常用于实现设备之间的低速数据通信。它广泛用于许多场合，特别是在通信领域内，以简单、方便、低速的特点用于数据链路层进行数据通信。

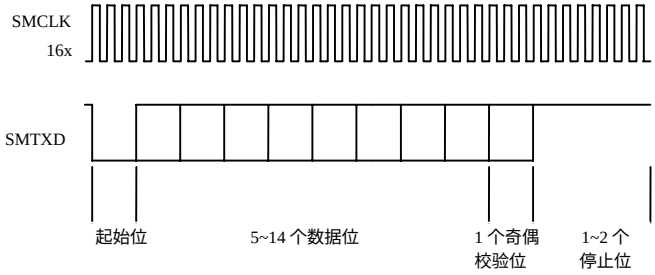


图1 SMC中的UART字符格式

UART协议是实现设备之间低速数据通信的标准协议。因发送时不需同时发送时钟，故此协议为异步。UART链接典型为38 400和9 600波特。SMC中的UART字符格式如图

基金项目： 国家“十五”“863”计划基金资助重大项目(2003AA12331004)

作者简介： 吴雨舟(1983－)，男，硕士研究生，主研方向：嵌入式系统开发；路唯佳，硕士研究生；张 平，博 士、教授、博士生导师

收稿日期： 2006-09-11 **E-mail：** wnyuzhou@gmail.com

1 所示，为 1 个起始位，5~14 个数据位，1 个奇偶位(可选)，1~2 个停止位。接收器、发送器异步工作，无需连接接收和发送时钟。接收器采取对输入数据流高度采样的方式，通常采样为 16，并根据采样值确定位值。按惯例，使用 16 个采样值的中间 3 个值。

SMC 的参数 RAM 中有一个参数 MAX-IDL(Maximum idle characters)，用来设置空闲字符的多少。一旦一个字符被接收，SMC 开始计数接收到的空闲字符。若下一个数据字符接收前，当 MAX-IDL 多个空闲字符被接收，则产生空闲时间，缓冲区被关闭。顺次对 CPU 核心发出中断请求，要求从缓冲区接收数据。因此，MAX-IDL 给 UART 模式提供区分帧的便利方法。空闲字符按以下公式计算其位数：1(起始)+数据长度(5～14)+1(若奇偶校验被使用)+停止位(1～2)。

3 驱动程序的设计和实现

3.1 SMC 的工作过程

SMC 是通过对发送/接收缓冲区描述符(TxBD/RxBD)的操作来完成缓冲区中数据的通信。缓冲区描述符(BD)存储在双端口 RAM 中，包括发送 BD(TxBD)和接收 BD(RxBD)。每个 BD 都包括状态、数据长度和缓冲区指针等 3 个参数。每个 BD 对应着一个缓冲区。通过 BD 的状态字来对缓冲区中的数据执行相应的操作。SMC 的内存结构如图 2 所示。

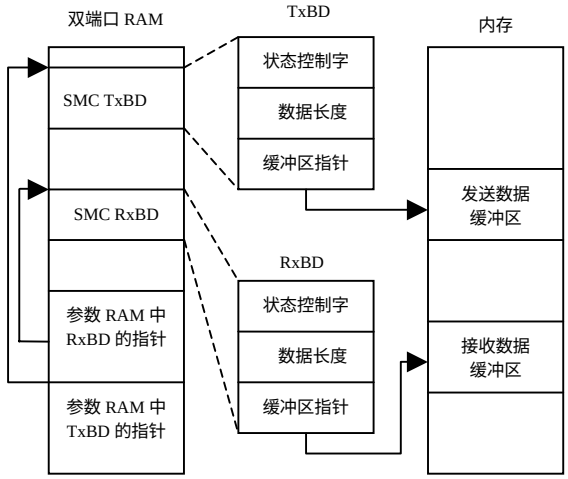


图 2 SMC 的内存结构

CPM通过SMC的参数RAM来获取RxBD/TxBD的基址、最大接收缓冲区长度以及SMC使用的通信协议信息等与SMC操作相关的参数或寄存器。参数RAM同样是存储在双端口RAM中的，其基址必须是关于 64 字节对齐的。此外，CPM中还包含一段用于参数RAM索引的内存空间。在这段内存中，FCC、SCC、SMC、SPI、I^2C、USB和IDMA等的参数RAM的首地址都存放在固定的偏移地址中，CPM会根据不同偏移地址找到相应的控制器参数RAM并对其进行读取。对于SMC而言，需要将参数RAM的首地址的值存入到相对双端口RAM偏移为 0x87FC的内存中。这样CPM就可以通过参数RAM索引找到SMC的参数RAM，进而对SMC进行相应操作。

除了以上所述的建立并初始化 BD 和参数 RAM，在发送和接收事件发生之前，还必须完成用于 SMC 收发的 CPM I/O 端口管脚的配置，CPM 包含 4 个通用目的 I/O 端口：PA，PB，PC，PD。端口的每个引脚都可用于通用目的 I/O 信号或外设接口信号。完成管脚配置后，需要为 SMC 配置波特率发生器为其提供内部时钟信号。最后还需要设置 SMC 模式寄存器 SMCMR，SMC 工作在 URAT 模式下，同时将 TEN、REN 比特位置 1 使能 SMC 的发送和接收。

当处于发送状态时，内核使能 SMC 发送，使 SMC 处于发送空闲状态。然后 SMC 对第 1 个 TxBD 进行轮询，如果有发送请求，SMC 将会从内存中读出数据进行发送。当 TxBD 数据将发送 FIFO 充满时，SMC 读取 TxBD 的状态字同时将状态字的 R 比特位清零，同时向 CPM 发中断请求。如果下一个 TxBD 也准备发送，则发出发送请求同时对其进行发送操作，若下一个 BD 没有准备发送，则 SMC 恢复到发送空闲状态等待下一个 TxBD 准备发送。

当处于接收状态时，SMC 等待第 1 个接收字符。如果第 1 个 RxBD 为空，则开始将接收字符存入缓冲区中。若接收缓冲区满或者接收到的空闲字符超过 MAX-IDL 所设置的最大空闲字符，则 SMC 清空状态字的 E 比特位同时发出中断请求。当接收数据超过缓冲区长度，SMC 将会读取下一个空闲的 RxBD，继续将接收数据存入此 RxBD 的缓冲区中。

3.2 驱动程序的启动流程及实现

驱动程序主要完成将 CPM 的 SMC1 配置成 UART 控制器来实现设备间通过串口的通信。驱动程序和操作系统是密切相关的，VxWorks 下的串口驱动程序是包括在板级支持包 BSP 内的。

VxWorks 的启动涉及到两个映像文件：BootRom 映像和 VxWorks 映像。BootRom 也使用 VxWorks 内核，它其实可以看成是一个最小化的 VxWorks 映像，只是对系统进行了最基本的初始化，以便能够最终加载 VxWorks 映像。一般串口驱动在 BootRom 映像中就必须被成功装载。系统上电后，首先处理器跳转到 ronInit.s 文件中的 romInit( )函数的首地址执行，主要完成屏蔽中断、初始化 CPU、初始化内存。之后跳转到 bootInit.c 文件中的 romStart( )中，执行必要的代码压缩和 ROM 型映像的重定位。完成后跳转执行 bootConfig.c 文件中的 usrInit( )。usrInit( )函数主要完成最基本的系统初始化，它会调用 sysLib.c 文件中的 sysHwInit()初始化与目标板相关的硬件，在这个函数中串口驱动程序的入口函数 sysSerialHwInit()将会被调用。

串口驱动程序包括 sysSerial.c、smc8260Sio.c 和 smc8260Sio.h文件。其中后两个文件是PQII系列处理器在VxWorks下通用的驱动程序文件。smc8260Sio.c用来实现基本的串口操作的功能，是SMC设备UART驱动的主体。在smc8260Sio.h中定义了SMC设备串行I/O通道的数据结构ppc8260smc_chan，其中包括配置SMC设备相关的寄存器以及SIO各驱动函数(smc8260Ioctl()、smc8260Startup()、smc8260CallbackInstall()、smc8260PollInput()、smc8260PollOutput())的定义。如果需要改变串口驱动程序的工作模式而不改变功能和收发BD数目的情况下只需要重新配置smc8260Sio.h中的SMC设备相关寄存器的值就可以实现。

sysSerial.c 是串口驱动的配置文件，在逻辑上属于 smc8260Sio.c 的上一级接口。该文件中的 sysSerialHwInit()通过调用 smc8260Sio.c 中的相关函数，将串口初始化到一个静止状态并向操作系统注册例程。之后 sysSerialHwInit2()将挂接串口中断并始能串口。

具体调用关系如图 3 所示，包括 sysSerialHwInit()和 sysSerialHwInit2()是如何调用 smc8260XXX()函数的。

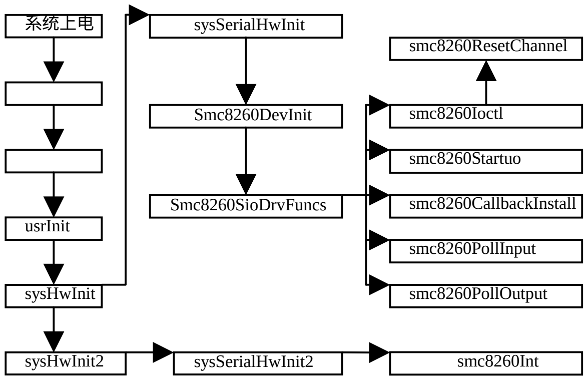


图 3　驱动程序调用流程

具体的函数功能以及实现如下：

sysSerialHwInit( )，在此函数中主要完成以下功能：配置管脚资源 PD9 用于输出，PD8 用于输入；配置波特率发生器 BRG7 和使用的波特率(在函数 smc8260ResetChannel( )中将 BRG7 设置为 SMC1 所使用的波特率发生器)；利用库函数 m82xxDpramAlignedMalloc 在双端口 RAM 中分配发送/接收 BD、发送/接收缓冲区以及 SMC1 参数 RAM 的内存区，同时将 SMC1 参数 RAM 的内存首地址存入双端口 RAM 偏移 87FC 的内存中。完成以上操作后调用 smc8260DevInit 函数初始化 SMC 设备。

smc8260DevInit( )，此函数将 SIU 中断屏蔽寄存器 SIMR_L 中的 SMC1 比特位置 0 屏蔽 SMC 中断，然后调用串口驱动程序管理函数 smc8260SioDrvFuncs( )，由此函数调用各个驱动程序实现函数。

smc8260Ioctl( )，根据不同要求实现设置或获得波特率，设置或获得工作模式。此外，该函数通过对 SIU 的中断控制寄存器 SIMR_L、SIPNR_L、SMCE、SMCM 等对应 SMC1 的比特位进行操作可以完成切换查询/中断模式。

smc8260ResetChannel( )，此函数是对 SMC1 通道进行初始化，包括设置 CMXSMR，将 SMC 配置成 NMSI 模式同时选择 SMC1 所使用的波特率发生器；建立 TxBD、RxBD 并初始化每个 BD，包括状态/控制字、数据长度、对应的缓冲区地址，同时初始化参数 RAM 中的参数；设置 SMC 模式寄存器 SMCMR，使 SMC 工作在 URAT 模式下，同时将 TEN、REN 比特位置 1，使能 SMC 发送和接收；设置 SMCE 清除 SMC 所有的中断及响应事件，同时设置 SMCM 屏蔽发送、接收中断。

smc8260Startup( )、smc8260PollInputsmc( )、8260 PollOutput( )等 3 个函数实现在检查 SMC 通道状态后，完成发送或接收。

sysSerialHwInit2( )，将 SMC1 与相应的中断处理函数进行连接，包括将中断服务函数 smc8260Int( )与 4 号中断连接，同时设置 SIMR_L 开中断。

smc8260Int( )，发生 SMC 中断时调用该函数处理中断，完成数据的接收或发送。

4　结束语

本串口驱动经过长时间的应用和测试，在整个系统中运行稳定。可以通过串口对整个系统及其外设进行配置，同时还可以监控 VxWorks 下运行的应用程序状态。

不仅仅是SMC，CPM的FCC、SCC、SPI、I^2C、USB和IDMA等控制器也都是通过BD及其参数RAM来完成数据的传输与通信，工作过程也与SMC有很多的相同之处。因此，在编写这些驱动程序时都可以参考SMC URAT 串口驱动程序的实现。

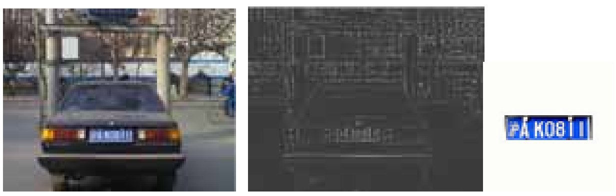
参考文献

1 Motorola. MPC8280 PowerQUICC II™ Family Reference Manual[Z]. 2004.

2 Wind River. VxWorks BSP Developer’s Guide 5.5[Z]. 2002.

3 Wind River. Tornado™ BSP Training Workshop[Z]. 2002.

（上接第 266 页）



(a)原图　　(b)细节分量图　　(c)分割结果

图 3　车牌图像的定位与分割

图 3(a)为一复杂背景的图像原图。图 3 (b)为对原图进行灰度化、归一化处理，以及 3 级小波分解和细节分量重构后，获得的小波变换细节分量图。图 3 (c)为基于亮度矩的车牌特征和先验知识定位车牌分割后的结果。由于原图上没有类似车牌区域，因此只分割下牌照。

由此可见，基于小波分解和亮度矩的车牌字符定位能较好地抑制干扰信号，突出字符区域特征，定位准确。

4　结束语

本文运用小波分解作为研究车牌图像牌照定位的数学工具，在定义了亮度矩函数的基础上，提出了基于小波和亮度矩的车牌定位方法。该方法对牌照区域大小、位置以及图像背景的限制小，具有容噪、抗光照不均匀和定位速度快的性能。对大量具有复杂背景的图像进行仿真研究显示，本文提出的车牌定位方法准确率达到 97%以上，同时适用于图书和广告等画面文字的定位与分割。

参考文献

1 张　引，潘云鹤. 彩色汽车图像牌照定位新方法[J]. 中国图像图形学报, 2001, 6(4): 374-377.

2 Bister M, Cornelis J, Rosenfeld A. A Critical View of Pyramid Segmentation Algorithms[J]. Pattern Recognition Letters, 1990, 11(9): 605-617.

3 Raus M, Kreft L. Reading Car License Plates by the Use of Artificial Neural Networks[C]//Proceedings of the 38th Midwest Symposium on Circuits and Systems. 1995: 538-541.

4 Nalwa V S. On Detecting Edges[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1986, 8(6): 699-714.

5 Sahoo P K. A Survey of Thresholding Techniques[J]. Computer Vision Graphics and Image Processing, 1988, 41(2): 233-260.