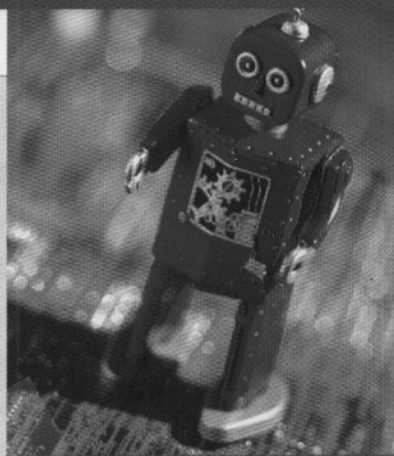


# 基于PXA250和Linux嵌入式系统在机器人遥操作中的设计与实现

## Design and Implementation of an Embedded System in Remote Robot Control Based on Linux



刘殿敏 李科杰

(北京理工大学机电工程学院 100081)

**摘要:** 本文介绍了基于 PXA250 和 Linux 嵌入式系统的硬件及软件设计与实现。描述了基于 PXA250 嵌入式系统硬件设计原理,探讨了 Linux 操作系统裁剪及生成适合嵌入式的 Linux 操作系统。给出了全双工串口通信和网络 socket 通信编程设计与实现。

**关键词:** PXA250 嵌入式 Socket 线程同步 串口通信 遥操作

### 1 引言

网络技术应用于遥操作系统,进一步拓宽了网络和现实世界相互作用的应用领域,Internet在数据共享和信息高速传输方面有强大的功能,使人感到遗憾的是目前多数基于Internet的应用还不能提供与真实世界的实时交互,更不用说直接实时地控制和改变周围环境。因此,和现有的Internet技术结合起来,尽可能的延伸Internet的应用领域。进行远程教学、远程医疗及远程作业等,具有深远的意义。嵌入式系统完成网络通信和数据采集处理的重要任务,是遥操作系统的重要组成部分,所以选用网络功能强大的Linux OS和运行速度较高的PXA250微处理器。

### 2 基于 PXA250 和 Linux 嵌入式系统在远程机器人遥操作中的作用

遥操作发出的控制指令通过无线局域网发送到嵌入式系统,嵌入式系统通过译码,把控制指令传递给机器人的控制系统,同时采集机器人的状态信息,如机器人前方的图像、环境信息、运动状态和位置信息等实时地通过无线局域网发送给遥操作控制显示系统,以便遥操作者,根据机器人的环境信息和运动状态,决定下一步的操作任务,系统原理如图1所示。

### 3 基于 PXA250 和 Linux 嵌入式系统的硬件设计

系统硬件原理示意图如图2所示,系统由Intel XScale PXA250

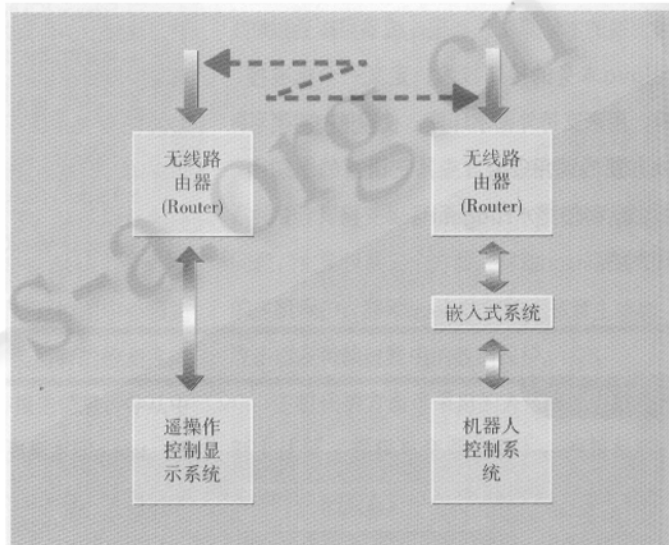


图1 遥操作原理示意

处理器、两片16M Byte E28F640的Flash、两片16M Byte K4S281632C的SDRAM、10bits 4通道的A/D转换芯片UCB1400、网络接口芯片CS8900、USB接口驱动芯片ISP1611和一片复杂可编程逻辑器件(CPLD) Altera EPM7128AETC100-7等主要器件组成。PXA250微处理器内集成有: SDRAM刷新和控制器; LCD驱动和控制器; 一个串行口和红外接口; 一个MMU控制器; 键盘接口; 一个USB(slave)接口。PXA250 处理器可运行于200/300/400MHz (Xscale core), 运行的速度可有软件设置, 核心电源的电压为0.80V至1.45V, 32Bit

RISC处理器,具有32K指令缓冲,32K数据缓冲,MMU单元,Minicache,很低的电源要求,可由锂电池或AA/AAA电池供电,内部有PCMCIA控制器(PXA210除外)。

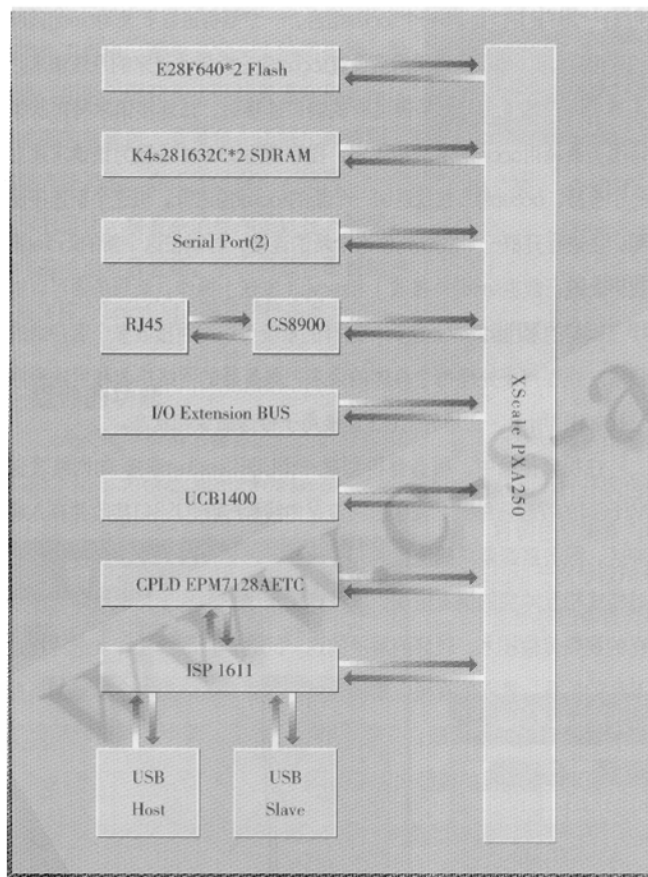


图2 嵌入式系统原理示意

## 4 基于PXA250和Linux嵌入式系统的软件设计

### 4.1 Linux操作系统的裁剪

嵌入式操作系统是整个嵌入式系统的核心,但典型Linux操作系统是为桌面配置的,内核十分庞大,而嵌入式系统的内核十分有限,因此,要把Linux操作系统装入有限的内存,就要对它进行裁剪,裁剪过程设计的主要技术有:内核的精简;虚拟内存机制的屏蔽和设备驱动程序编写。内核的精简,标准Linux是面向PC,集成了许多PC需要而嵌入式系统不需要的功能,对一些可独立加上和卸下的功能模块,可在编译内核时,仅保留嵌入式系统所需的功能支持模块,删除不需要的功能模块。虚拟内存机制的屏蔽,实现虚拟内存的机制有:地址映射机制、内存分配和回收机制、缓存和刷新机制、清页机制、交换机制、内存共享机制,将实现这些机制的数据结构和函数进行修改或屏蔽,还要修改与之相关的文件。确定了内核的基本功能后,就要为特定的设备编写驱动程序,可按照Linux编写驱动程序的规则编写,编写

的驱动程序应具有的功能:对设备进行初始化和释放、把数据从内核传到硬件和从硬件读取数据、读取应用程序传递给设备文件的数据和回送应用程序请求的数据、检测和处理设备出现的错误。一个小型的Linux操作系统完成后,它包括进程管理、内存管理和文件管理,支持多任务并行;基于闪存的文件系统,应用程序和重要的数据以文件的形式被放在闪存系统中;有完整的TCP/IP协议栈,Linux内建有对以太网控制器的支持;有完整的USB协议栈和驱动。还要将剪裁好的Linux内核移植到目标板上,在通常移植内核时,首先将内核编译成针对该处理器的目标代码,而本系统采用的是Intel公司的PXA250处理器,它有一些不同与其他CPU的地方,一些内核程序需修改,涉及到编写Linux的引导代码和修改与体系结构相关部分的代码。将Flash作为系统启动设备,引导代码放在Flash上。系统加电后,有引导代码进行基本硬件的初始化,然后把内核影象装入内存运行。

### 4.2 全双工串口通信程序的设计

Linux串口的资源,Linux所有的硬件资源都虚拟成文件,通过虚拟文件统一管理硬件设备。内核提供访问设备文件的系统函数,应用程序通过调用这些函数完成对硬件的操作。访问一个串口,只需要打开相应的设备文件,然后向这个文件读写数据就可以完成数据的接收和发送。Linux串口的配置,串口所有的配置选项均存放在termios的结构中,termios是一个复杂的数据结构,包括波特率(Baudrate)、数据位(Bits)、停止位(Stop)及校验方式(Parity)等,并且包含很多不常用的参数,对串口进行配置,首先读取当前串口的termios;根据要求改写termios,然后把改写的termios重新配置串口;在程序结束时,恢复系统原来的termios。串口类的实现,为了有效的解决串口通信实时响应和降低数据的丢失率,采用串口全双工多线程类的框架。基于多线程技术可方便的实现应用程序的多任务,线程的同步采用互斥同步,对于全双工串口通信,程序的输入和输出时两个需要并发执行的操作,因此必须多线程技术,在这里笔者在串口接收和发送分别创建的两个线程分别为线程A和线程B,而网络的接收和发送线程分别为线程C和线程D。发送线程B把网络接收线程C接收并且存储在buffer2的信令数据,经过译码后,发送给控制系统的主机,接送线程A完成接收从控制系统发送机器人的环境状态信息,存入缓冲区buffer1,然后网络发送线程D把buffer1的数据通过网络传到遥操作控制前端。考虑到读缓冲区的过程不能写,写缓冲区的过程不能读,因此要采用线程同步,采用由互斥锁来控制,实现线程之间同步,线程A和线程C,线程B线程D分别由互斥锁control\_mutex来控制。下面分别为接收和发送线程,头文件定义如下:

```
#include<pthread.h>
#include<thread_extensions.h>
#include<termios.h>
```

```

pthread_mutex_t count_r_mutex=PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t count_s_mutex=PTHREAD_MUTEX_INITIALIZER;
int buffer1_full=0,buffer1_len;
int buffer2_full=0,buffer2_len;
char buffer1[buffer1_len]
char buffer2[buffer2_len]
//串口接收函数的定义
void recieve(fd,r_buffer,nszie)
{
while(1){
pthread_mutex_lock(& count_r_mutex);
if(buffer1_full==0){
nszie=read(fd, buffer1,buffer1_len);
buffer1_full=1;}
pthread_mutex_unlock(& count_r_mutex);
}}
//串口发送函数的定义
void send(fd,s_buffer,nszie)
{
while(1){
pthread_mutex_lock(& count_s_mutex);
if(buffer2_full==1){
nszie=write(fd, buffer2,buffer2_len);
buffer2_full=0;}
pthread_mutex_unlock(& count_s_mutex);
}}
main(int argc,char **argv)
{
pthread_t thread_r_id,thread_s_id;
pthread_create(&thread_r_id,NULL,&recive,NULL);
pthread_create(&thread_s_id,NULL,&recive,NULL);
..... }

```

串口接收和发送线程也可以用C++编写，QThread是QT库的多线程类，从这个类派生出一个新类，并实现Qthread中的纯虚函数run()，即完成一个子线程的创建。

#### 4.3 网络通信程序的设计

由于采用Linux操作系统，它支持TCP/IP网络传输协议，网络数据传送分为命令和图像数据传送，命令传送通道采用TCP套接口的通信方式（面向连接的数据传送）虽然需要对差错进行控制和重新传送，但系统信令消息短，传送量小，因此可以承受所增加的额外开销。图像数

据传送是采用UDP的套接口通信方式（无连接的数据报传送），这是因为UDP通信不需要验证数据的正确接收，通信速度快，系统开销小，适合于容许少量数据丢失的视频流传输。服务器的主进程开始运行时，主进程就创建一个套接口，首先由socket()创建一个套接口；其次要完成一个规定的地址与一个socket结合，一个服务进程要使用bind()调用实现绑定；listen()系统调用用来设置socket使它能接收进来的系统请求；当某个客户端试图与服务器监听的端口连接时，该连接请求将排队等待服务器调用accept()函数接收它，程序调用accept()函数为该请求建立一个连接，accept()函数就返回一个新的socket描述符，供这个新连接使用，主进程开辟一个线程处理这个新的连接，可以在这个新的socket描述符上进行数据send()（接收）和recv()（发送）操作，而服务器可以继续以前的那个socket上监听，这样的系统可以同时接收多个客户端的请求。线程用于处理每个具体的请求，服务器的网络接收和发送程序流程如图3所示。它是一个TCP连接用于传送命令和状态。

在视频传输中，服务器再创建一个UDP socket连接，向网络发送MPEG4的视频压缩流，视频压缩流是由线程F通过USB口接收并存入缓冲区，服务器端要求码流发送和接收互不干扰，避免发送和接收相互等待或相互冲突的情况出现，本设计采用了一种基于环形缓冲区多线程的视频传输技术，较好的解决这一问题，读取线程和发送线程是互斥的关系，读入的视频数据供传输线程处理，这是一个简单的消费者和生产者的关系问题，只要不进入临界区，两个线程可以同时工作。UDP连接编程的模型如图4所示。

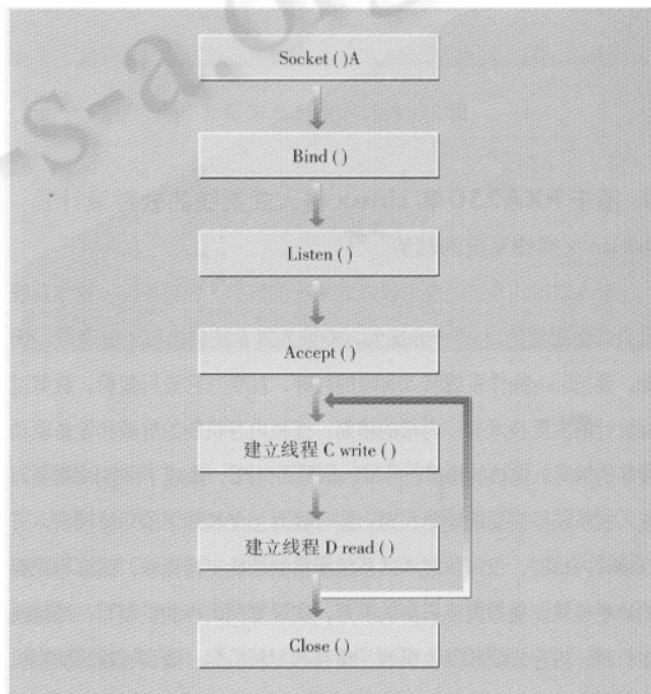


图3 面向TCP连接编程的模型

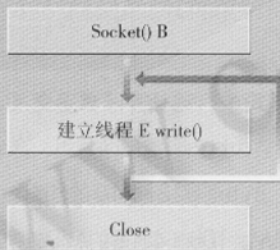


图 4 面向 UDP 连接编程

## 5 结论

Linux操作系统由于它的可靠性和强大的网络功能，正在被广泛地应用于各种系统，在设计一些复杂的网络产品，使研发周期大大缩短。

PXA250由于它的低功耗，内部集成一些外围接口模块，使硬件设计简单，系统稳定可靠，PXA250处理器可运行达到400M，所以很适合开发网络和控制系统的某些高端产品。本系统采用Linux操作系统和功能强大的PXA250处理器，能实时和可靠地通过无线局域网传送信息和指令，遥操作者通过遥操作系统能准确无误地控制远端的机器人完成一些复杂的任务，同时遥操作者可以感受机器人和环境的相互作用，有临场感。

## 参考文献

- 1 王学龙编著，嵌入式Linux系统设计和应用，清华大学出版社，2001。
- 2 邹思秩编著，Linux嵌入式设计与应用，清华大学出版社，2002。
- 3 David A.Rusling. The Linux Kernel. Version 0.8-3,1999。
- 4 K.Wall,M.Watson and M.Whitis.Linux Programming Unleashed,1999。