DOI:10.16660/j.cnki.1674-098x.2008.24.026

基于S3C2410的串口设计及其在Linux下编程

杨永金
(厦门海洋职业技术学院 福建厦门 361012)

摘 要:结合设计出来的S3C2410的串口通信电路,介绍了嵌入式系统在Linux下进行串口通信程序的开发,实现了嵌入式系统和PC机的串口通信。该程序利用Linux系统调用,开发简单、可移植性强、有利于大型程序的开发。
关键词:嵌入式 串口 Linux
中图分类号:TN91 文献标识码:A 文章编号:1674-098X(2008)08(c)-0042-02

1 前言

嵌入式系统广泛地应用于工业控制、数据采集、数控、自动化生产设备、汽车电子、智能仪表、通讯等领域。随着国内各种嵌入式产品的进一步开发和推广,嵌入式系统应用技术和人们的生活结合越来越紧密。由于嵌入式系统软件不像微机软件那样可以直接在本机上开发与调试,而是在微机上建立一个嵌入式的交叉编译环境,在此基础上进行软件的开发与调试,在编译调试过程中常使用串口通信的方式进行。而在实际应用中,有时需要借助微机的强大的数据处理能力和丰富的软件资源,使得组成的系统更为强大。为了提升应用系统的整体性能,必须实现PC机和嵌入式系统的通信。由于串行通信相对于并行通讯有连接线数量少、抗干扰性能好等优点,一般在实现PC机和嵌入式系统的通信中都使用串行通信方式。故嵌入式系统与本地PC机通信的主要方式,串口通信是必不可少的。本文将介绍S3C2410嵌入式系统上串行通信模块的设计及在移植了Linux的嵌入式系统上进行串口应用的开发。

2 系统硬件结构

2.1 系统框架

S3C2410芯片有3个UART(用于串行接收和传送)接口,这些接口用于支持串行异步通信。利用S3C2410上的这些UART接口,可以自己设计串行通信接口与其它设备进行通信。这里我们以嵌入式与PC机串行通信为例来介绍。

在嵌入式系统开发中,采用的是宿主机与目标机之间进行的远程调试,通常将运行目标程序的计算机系统称为目标机。由于嵌入式环境的目标机常常没有完善的人机接口,因此,需要另外一台通用计算机来辅助进行调试,这台运行高度环境的通用计算机通常称为宿主机。在目标机和宿主机之间需要通过一定的信通进行通信,完成调试信息的传递。一个常规的嵌入式系统的高度系统包括:宿主机、目标机、通信信道三部分。通信信道可以是并行接口、串行接口或以太网接口。采用串行接口通信的系统框架如图1。

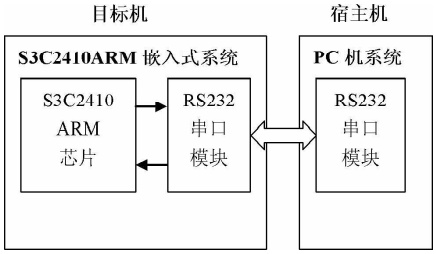

图1 系统框图

2.2 RS-232接口电路

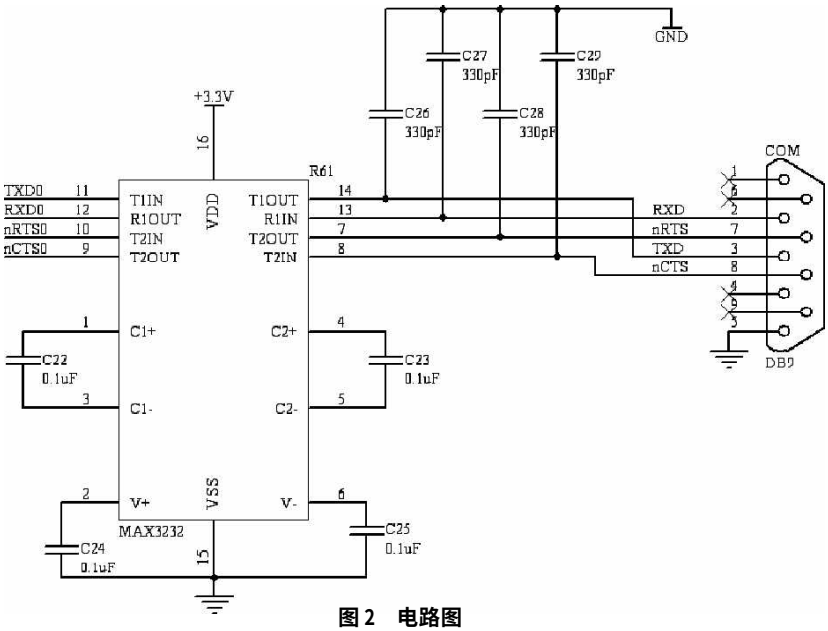

图2 电路图

嵌入式系统中,RS-232接口是广泛使用的串行通信接口。因为,RS-232C标准中表示逻辑0、1状态的电平,与嵌入式微处理器表示逻辑0、1状态的电平不同。因此为了使嵌入式微处理器能与RS-232C接口的设备连接,必须在嵌入式系统的RS-232C接口中设计电平转换电路。实现这种转换的集成电路芯片有多种,目前较广泛使用的芯片有MC1488、SN75150芯片(实现TTL电平到EIA电平的转换),MC1489、SN75154芯片(实现EIA电平到TTL电平的转换),MAX232芯片(实现TTL与EIA双向电平转换)。这里我们要实现全双工方式的串行通信,故采用MAX3232芯片。具体电路如图2。

S3C2410芯片的UART0相关引脚(即TXD0、RXD0、nRTS0、nCTS0)经过MAX3232电平转换后连接到DB9型的插座上。这样就可以使用S3C2410芯片内部的UART0部件来控制符合RS-232标准的串行通信。对于近距离的RS-232通信接口来说,通常不需要数据通信设备(即调制解调器),因此其接口只需要连接TXD和RXD信号线,nRTS和nCTS信号线可以不连接。但应注意,若使用近距离RS-232通信时,2台数据终端设备之间的通信电缆插座应交叉连接,即一端的RXD通过电缆与另一端的TXD连接,两端的“地”通过电缆连接在一起。

3 软件设计流程

由于在嵌入式系统上已经已经移植了Linux操作系统,对串口的编程工作就省去了用汇编语言或C51自己编写设备的初始化以及读写访问程序这样复杂的工作,且这样也不利于大规模的开发和设计。

Linux操作系统把串口及其他外设都当成普通文件进行操作,读写方便,因此进行相应的开发可以大大提高系统编程效率、简化调试的复杂程度。在Linux中,所有的设备文件一般都位于“/dev”下,其中串口一、串口二对应的设备名次依次为“/dev/ttyS0”、“/dev/ttyS1”,可以查看在“/dev”下的文件以确认,在这里S3C2410使用UART0,其对应的文件为“/dev/ttyS0”,由于在Linux下对设备的操作方法与对文件的操作方法是一样的,因此,对串口的读写可以使用简单的“read”,“write”

函数来完成，所不同的只是需要对串口的其他参数另做配置，串口读/写程序流程图如图3。

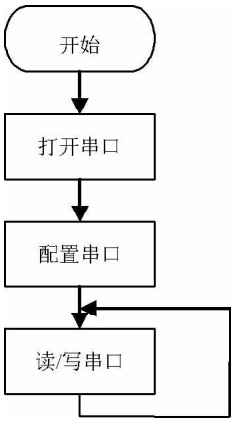

图3 串口程序流程

(1)打开串口

打开串口和打开普通文件一样，使用的函数同打开普通文件一样，都是open函数。如下所示：

```
Fd=open("/dev/ttyS0",O_RDWR|O_NOCTTY|O_NDELAY);
```

(2)串口详细配置

使用串口之前必须设置相关配置，包括：波特率、数据位、校验位、停止位等。串口设置由下面结构体实现：

```
Struct termios {
    tcflag_t  c_iflag;      /* input flags */
    tcflag_t  c_oflag;      /* output flags*/
    tcflag_t  c_cflag;      /*control flags */
    tcflag_t  c_lflag;      /* local flags */
    tcflag_t  c_cc[NCSS];   /* control characters */
}
```

该结构中c_cflag最为重要，可设置波特率、数据位、校验位、停止位。在设置波特率时需在数字前加上‘B’，如B9600、B19200。使用其需通过“与”“或”操作方式。

输入模式c_iflag成员控制端口接收端的字符输入处理。

常用串口控制函数如表1所示。

串口配置流程如下：

A.保存原先串口配置，使用tcgetattr(fd,&oldtio)函数

```
struct termios newtio,oldtio;
tcgetattr( fd,&oldtio );
```

B.激活选项有CLOCAL和CREAD，用于本地连接和接收使能。

```
newtio.c_cflag |=CLOCAL | CREAD;
```

C.设置波特率，使用函数cfsetispeed、cfsetospeed

表1 串口控制函数

Tcgetattr	取属性（termios结构）	Cfsetispeed	设置输入速度
Tcsetattr	设置属性（termios结构）	Cfsetospeed	设置输出速度
Cfgetispeed	得到输入速度	Tcdrain	得到前台进程组ID
Cfgetospeed	得到输出速度	Tcsendbreak	设置前台进程组ID

```
cfsetispeed(&newtio,B115200);
cfsetospeed(&newtio,B115200);
```

D.设置数据位，需使用掩码设置。

```
newtio.c_cflag &=~CSIZE;
newtio.c_cflag |=CS8;
```

E.设置奇偶校验位，使用c_cflag和c_iflag。

```
设置奇校验：
newtio.c_cflag |=PARENB;
newtio.c_cflag |=PARODD;
newtio.c_iflag |=(INPCK | ISTRIP);
设置偶校验：
newtio.c_iflag |=(INPCK | ISTRIP);
newtio.c_cflag |=PARENB;
newtio.c_cflag &=~PARODD;
```

F.设置停止位，通过激活c_cflag中的CSTOPB实现。若停止位为1，则清除CSTOPB，若停止位为2，则激活CSTOPB。

```
newtio.c_cflag &=~CSTOPB;
```

G.设置最少字符和等待时间，对于接收字符和等待时间没有特别要求时，可设为0。

```
newtio.c_cc[VTIME]=0;
newtio.c_cc[VMIN]=0;
```

H.处理要写入的引用对象

tcflush函数刷清(抛弃)输入缓存(终端驱动程序已接收到，但用户程序尚未读)或输出缓存(用户程序已经写，但尚未发送)。

```
int tcflush(int filedes,int queue )
```

queue数应当是下列三个常数之一：

TCIFLUSH 刷清输入队列。

TCOFLUSH 刷清输出队列。

TCIOFLUSH 刷清输入、输出队列。

如:tcflush(fd,TCIFLUSH);

I.激活配置。在完成配置后，需激活配置使其生效。使用tsettattr()函数。原型：

```
int tcgetattr(int filedes,struct termios * termptr);
int tcsetattr(int filedes,int opt,const struct termios * termptr);
```

tcsetattr的参数opt使我们可以指定在什么时候新的终端属性才起作用。opt可以指定为下列常数中的一个：

TCSANOW 更改立即发生。

TCSADRAIN 发送了所有输出后更改才发生。若更改输出参数则应使用此选择项。

TCSAFLUSH 发送了所有输出后更改才发生。更进一步，在更改发生时未读的所有输入数据都被删除(刷清)使用如：tcsetattr(fd,TCSANOW,&newtio)

(3)读写串口

串口的读写与普通文件一样，使用read,write函数。

```
read(fd,buff,8);
write(fd,buff,8);
```

程序写好之后，将该程序进行交叉编译，得到目标机的可执行文件，然后将其下载或烧写到目标机上执行，就可以和目标机上的外围设备进行串行通信了。这里我们与宿主机进行通信发现程序运行稳定可靠。此时的宿主机可以运行是Linux下用相同方法编写的串口通信程序，也可以是运行Windows下串口调试程序。

4 结语

本文介绍了S3C2410嵌入式系统的串口通信电路的设计及在Linux下串口程序的开发。其中电路及软件都在实验中顺利的完成调试。串口通信程序的开发是通过Linux的系统调用完成的，所以该程序也可以运行在其它Linux操作系统下的嵌入式系统及PC机上。也可以被其它程序调用来完成相应的功能，有利于大型程序的开发。

参考文献

[1] 孙琼.嵌入式Linux应用程序开发详解[M].人发邮电出版社.
[2] 符意德,陆阳.嵌入式系统原理及接口技术[M].清华大学出版社.
[3] 杜春雷.ARM体系结构与编程[M].清华大学出版社.