

Linux操作系统下的串口通讯编程 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/206/2021_2022_Linux_E6_93_8D_E4_BD_c103_206474.htm

Linux 操作系统从一开始就对串口提供了很好的支持，本文就 Linux 下的串口通讯编程进行简单的介绍。串口简介 串口是计算机一种常用的接口，具有连接线少，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。它的全名是“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 个脚的 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。传输距离在码元畸变小于 4% 的情况下，传输电缆长度应为 50 英尺。Linux 操作系统从一开始就对串口提供了很好的支持，本文就 Linux 下的串口通讯编程进行简单的介绍，如果要非常深入了解，建议看看本文所参考的《Serial Programming Guide for POSIX Operating Systems》

序号	信号名称	符号	流向	功能
2	发送数据	TXD	DTE → DCE	DTE 发送串行数据
3	接收数据	RXD	DCE → DTE	DTE 接收串行数据
4	请求发送	RTS	DTE → DCE	DTE 请求 DCE 将线路切换到发送方式
5	允许发送	CTS	DCE → DTE	DCE 告诉 DTE 线路已接通可以发送数据
6	数据设备准备好	DSR	DTE → DCE	DCE 准备好
7	信号地			信号公共地
8	载波检测	DCD	DTE → DCE	DCE 表示 DCE 接收到远程载波
20	数据终端准备好	DTR	DTE → DCE	DTE 准备好
22	振铃指示	RI		

DTE DCE 表示 DCE 与线路接通，出现振铃串口操作串口操作需要的头文件#include /*标准输入输出定义*/#include /*标准函数库定义*/#include /*Unix 标准函数定义*/#include #include #include /*文件控制定义*/#include /*PPSIX 终端控制定义*/#include /*错误号定义*/

打开串口在 Linux 下串口文件是位于 /dev 下的串口一为 /dev/ttyS0串口二为 /dev/ttyS1打开串口是通过使用标准的文件打开函数操作：int fd./*以读写方式打开串口*/fd = open("/dev/ttyS0", O_RDWR).if (-1 == fd){ /* 不能打开串口一*/ perror(" 提示错误！").}设置串口最基本的设置串口包括波特率设置，效验位和停止位设置。串口的设置主要是设置 struct termios 结构体的各成员值。struct termio{ unsigned short c_iflag. /* 输入模式标志 */ unsigned short c_oflag. /* 输出模式标志 */ unsigned short c_cflag. /* 控制模式标志*/ unsigned short c_lflag. /* local mode flags */ unsigned char c_line. /* line discipline */ unsigned char c_cc[NCC]. /* control characters */}.设置这个结构体很复杂，我这里就只说说常见的一些设置：波特率设置下面是修改波特率的代码：struct termios Opt.tcgetattr(fd, amp.Opt,B19200). /*设置

为19200Bps*/cfsetospeed(amp.Opt).设置波特率的例子函数：/**@brief 设置串口通信速率*@param fd 类型 int 打开串口的文件句柄*@param speed 类型 int 串口速度*@return void*/int speed_arr[] = { B38400, B19200, B9600, B4800, B2400, B1200, B300, B38400, B19200, B9600, B4800, B2400, B1200, B300, }.int name_arr[] = {38400, 19200, 9600, 4800, 2400, 1200, 300, 38400, 19200, 9600, 4800, 2400, 1200, 300, }.void set_speed(int fd, int speed){ int i. int status. struct termios Opt. tcgetattr(fd, amp.Opt,

```
speed_arr[i]). cfsetospeed(&Opt). if (status != 0) {  
perror("tcsetattr fd1"). return. } tcflush(fd,TCIOFLUSH). } } }效验  
位和停止位的设置：无效验 8位 Option.c_cflag &amp;=   
~CSTOPB.Option.c_cflag &amp;= ~PARODD.Option.c_cflag  
&amp;= ~CSIZE.Option.c_cflag |= ~CS7.偶效验(Even) 7位  
Option.c_cflag &amp;= ~CSTOPB.Option.c_cflag &amp;=   
~PARENB.Option.c_cflag &amp;= &amp;~CSIZE.Option.c_cflag |=  
CS8. 100Test 下载频道开通，各类考试题目直接下载。详细请  
访问 www.100test.com
```