

# 485 通讯编程

## 概述

PCM338x 的 COM2 可通过 J2 跳线设定 232 或 485 ,当设为 485 时 J8 为 485 接口。详细的设置和 J8 的接口定义见手册 :c:\lanry\doc\产品手册\PCM-338x 用户手册.pdf。例子演示了如何连接 485 和对 485 编程。

## 演示环境

- 建立虚拟显示

虚拟显示用来使用户开发没有标准显示的嵌入式应用时，感觉就像在一台完整的 PC 机上进行开发。使用一台主机模拟了标准输入（键盘）和输出（屏幕）。

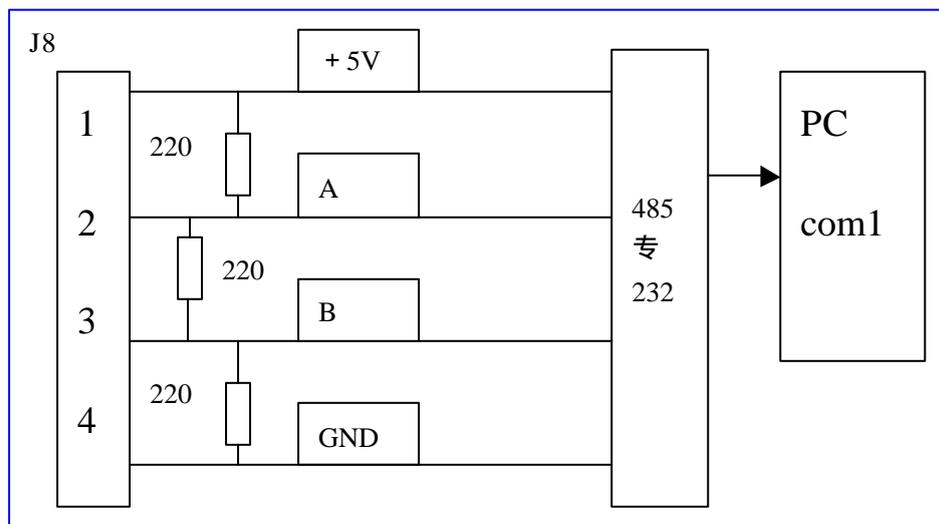
建立虚拟显示需要一台有标准串口 COM1(地址 3F8)的主机并运行纯 DOS 系统，用调试电缆一端连接目标板虚拟显示口，PCM3386 为 COM4，DVK2386 为 COM3，调试电缆另一端连接主机的 COM1；目标板打开虚拟现实跳线；关闭主机键盘的数字灯；主机运行虚拟显示软件 PCVID3V0.EXE。此时就可以看到目标板的显示了，也可以用键盘进行操作。如果连接我公司提供的液晶屏，在液晶屏上也有相同的输出。

参考文档：c:\lanry\doc\开发工具使用\虚拟显示使用方法.pdf

- 硬件设置与连接

PCM3386 和 PCM3388 的 COM2 可以通过 JP2 选择 232/485 电平。设置情况和接口信号定义请参考硬件手册：c:\lanry\doc\产品手册\PCM-338x 用户手册.pdf 和 c:\lanry\doc\硬件文档 PCM3386\_6v1 接口定义.pdf。

硬件电缆连接如下图所示：



PC 侧运行串口调试助手，波特率设为 115200，数据位为 8，无效验位，停止位为 1。

## 演示方法和效果

- 直接运行

在虚拟显示端键入

c:\example\ser\ser485.exe

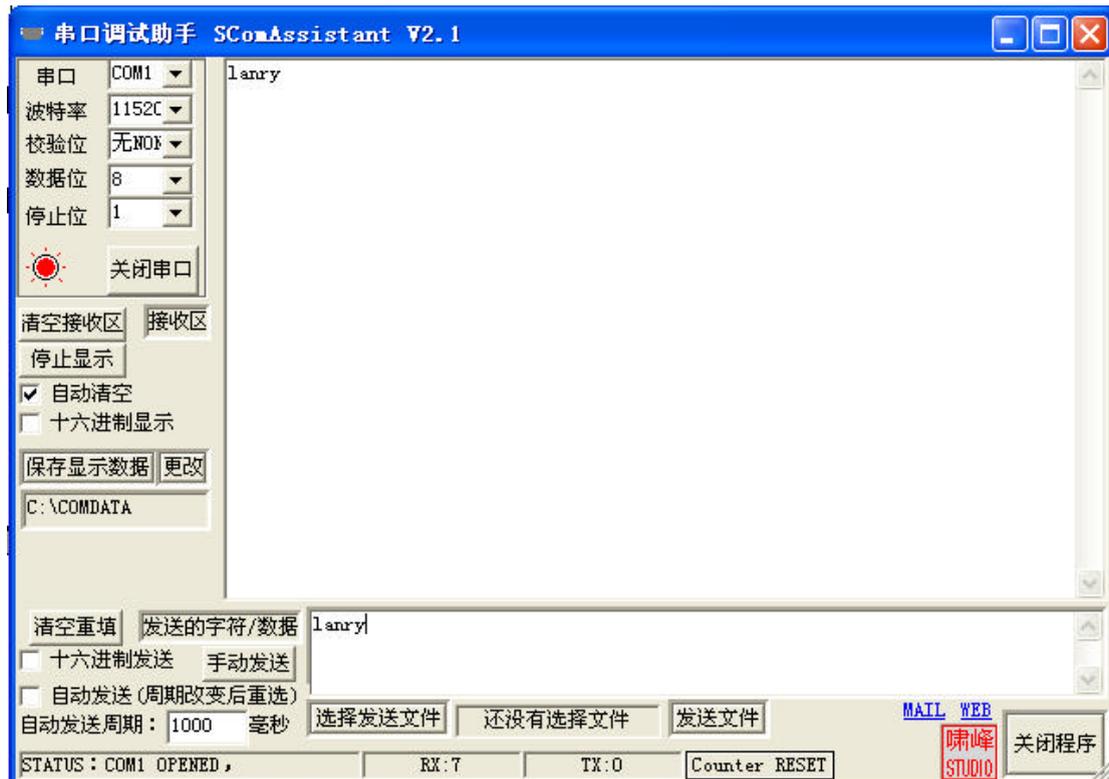
PC 侧运行串口调试助手，波特率：115200，数据位：8，效验位：无

NONE，停止位：1

在虚拟键盘上输入

lanry（注：在虚拟显示屏上无该字符显示）

PC 侧应显示如下：



在 PC 侧输入 lanry 并点击手动发送，虚拟显示端等屏幕刷新后应显示 lanry。

在虚拟显示端按“ESC”键退出。

- TD 下载运行

用户可以通过 TD 远程调试方式运行，用空猫电缆连接目标机的 COM1 和 PC 端的 COM1，目标端运行 tdreomte，通过 bc45 开发环境中的远程调试功能下载（tools - ）turbo debugger）。下载完毕在

TD 下按 F9 全速运行或按 F8 单步运行。该调试环境的建立参考文档：  
C:\lanry\doc\开发工具使用\使用 BC4.5 开发流程.pdf 和演示录像  
C:\lanry\doc\开发工具使用\用 BC45 开发调试程序演示录像.exe。

## 代码分析

485 为单工方式，485 的发送/接收模式通过 rts 信号控制。  
Ser485 中接收为中断方式，上电默认为 485 接收方式，发送时通过  
rts 信号控制方向。参见 ser485.c 源文件中的 int SerSend(char x)。

```
int SerSend(char x )
{
    long int          timeout = 0x0000FFFFL;
    unsigned char     MCRTemp;
    outportb(portbase + MCR,  MC_INT | DTR | RTS);      //设置MCR的RTS位，进入发送模式

    /* 硬件握手，如果用三线制时，需要把串口接成硬件握手自环方式或者把下面三句注释掉。*/
    while ((inportb(portbase + MSR) & CTS) == 0)
        if (!(--timeout))
            return -1;

    timeout = 0x0000FFFFL;

    /* Wait for transmitter to clear */
    while ((inportb(portbase + LSR) & XMTRDY) == 0)
        if (!(--timeout))
            return -1;

    disable();
    outportb(portbase + TXR, x);
    enable();

    timeout = 0x0000FFFFL;

    /* Wait for transmitter to clear */
    while ((inportb(portbase + LSR) & XMTRSR) == 0)
        if (!(--timeout))
            return -1;

    MCRTemp = inportb(portbase + MCR);                //清除MCR中的RTS位，使485处于接收状态
    MCRTemp = MCRTemp & ~RTS;
    outportb(portbase + MCR,  MCRTemp);
    return 0;
}
```