



UT-7708

PCI转8口RS-232多串口卡

说明书

一、概述

UT-7708是一款PCI转8口RS-232多串口卡，串口卡兼容RS-232标准，能够将单端的PCI信号转换为RS-232信号，带有快速的瞬态电压抑制保护器，此保护器被设计用来保护RS-232接口，采用当今先进的TVS(TRANSIENT VOLTAGE SUPPRESSOR)瞬态电压抑制器。

正常情况下，TVS管呈高阻状态，当TVS管两端经受瞬间的高能量冲击时，它能以极高的速度将其两端的阻抗降低，吸收一个大电流，从而把其两端的电压钳制在一个预定的数值上，保护后面的电路元件不因瞬态高压冲击而损坏。此保护器可以有效地抑制闪电(LIGHTNING)和ESD，且极小的极间电容保证了RS-232接口的高速传输。

二、产品特点

- 连接类型：DR62母头1个
- 总线类型：PCI
- 信号类型：RS-232：DCD、RXD、TXD、DTR、GND、DSR、RTS、CTS
- 传输速率：300bps-115.2Kbps
- 数据位：5、6、7、8
- 停止位：1、1.5、2
- 校验位：None、Even、odd、Space、Mark
- 流控制：RTS/CTS XON/XOFF
- 工作温度：-40℃ ~ +85℃
- 接口保护：RS-232接口每线差模 ±1KV雷击浪涌防护
- RS-232接口接触 ±4KV ESD防护、空气 ±4KV ESD防护
- 支持系统：windows xp/win7、8、8.1、10、11的32、64位等windows系统以及Linux系统

三、连接器和信号

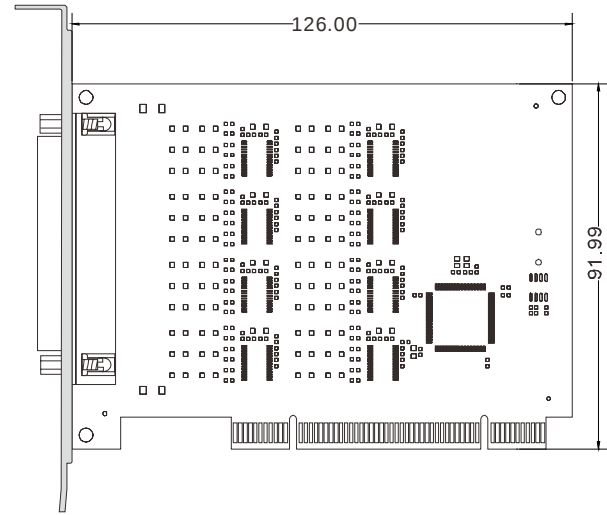
1、DB9针型：RS-232输出信号引脚分配（Port1-Port8）

DB9针型 (PIN)	RS-232接口信号
1	载波检测DCD
2	接收数据RXD
3	发送数据TXD
4	数据终端准备DTR
5	信号线GND
6	数据终端准备DSR
7	请求发送RTS
8	清除发送CTS
9	空

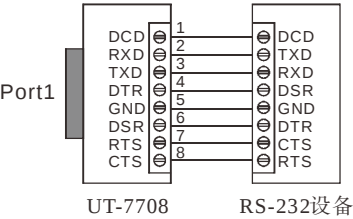


四、产品外形图

- 1、单位：mm
- 2、该板卡适用于4U机箱



五、通信连接示意图



六、故障及排除

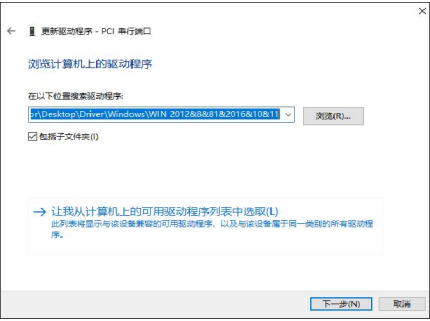
- 1、数据通信失败
 - A、检查PCI接口是否接好
 - B、检查RS-232输出接口接线是否正确
 - C、检查接线端子是否连接良好
 - D、检查供电是否正常
- 2、数据丢失或错误
 - A、检查数据通信设备两端数据速率、格式是否一致

七、安装驱动程序步骤（以windows10系统举例）

当插上UT-7708产品后，打开设备管理器，右键点击未识别的PCI串行端口，选择更新驱动程序软件，选择第二项“浏览我的计算机以查驱动程序软件”。



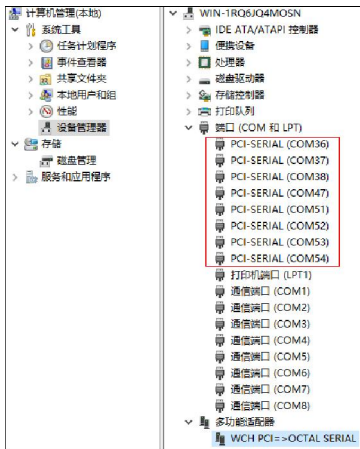
浏览文件位置，选择好对应系统的驱动文件夹，勾选“包括子文件夹”，并点击“下一步”。



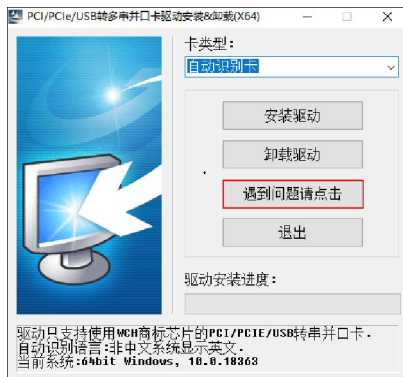
系统会自动选择文件夹内合适的驱动并安装上WCH PCI=>OCTAL SERIAL，点击“关闭”。



此时，串口驱动安装完毕，端口处显示所需的8个串口（如红色框选部分），每台电脑显示的串口号可能不同，以实际情况为准。



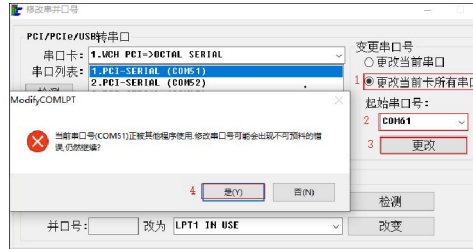
此时，串口号未按顺序查找，需要进一步对串口号进行排序。
打开驱动文件夹Driver，点击Windows，点击PUMPSETUP64
将出现以下界面。随后点击“遇到问题请点击”。



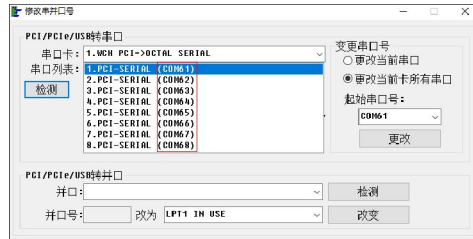
点击“修改串并口号”。



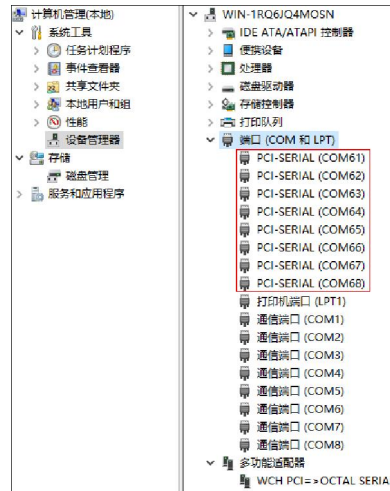
点击“更改当前卡所有串口”，选择“起始串口号”（以COM61为例），点击“更改”；ModifyCOMLPT界面点击“是”（后续出现的ModifyCOMLPT都点击“是”）



此时，串口排序完成。



重启电脑，打开设备管理器，串口排序完成，可以进行正常通信了。



八、Linux安装驱动程序步骤(以Ubuntu 18.04.5系统举例)

从终端内进入到Linux文件夹路径下，键入sudo sh install.sh命令，输入系统密码，出现以下界面，表示Linux驱动安装成功。

```
ub18@ub18:~/Driver/Linux$ sudo sh install.sh
[sudo] ub18 的密码:
make -C /lib/modules/5.4.0-150-generic/build M=/home/ub18/Driver/Linux/driver_pre modules
make[1]: 进入目录"/usr/src/linux-headers-5.4.0-150-generic"
Building modules, stage 2.
MODPOST 1 modules
make[1]: 离开目录"/usr/src/linux-headers-5.4.0-150-generic"
lsmod wch_pre.ko || true
mkdir -p /lib/modules/5.4.0-150-generic/kernel/drivers/tty/serial || true
cp -f ./wch_pre.ko /lib/modules/5.4.0-150-generic/kernel/drivers/tty/serial || true
depmod -a
make -C /lib/modules/5.4.0-150-generic/build M=/home/ub18/Driver/Linux/driver_main modules
make[1]: 进入目录"/usr/src/linux-headers-5.4.0-150-generic"
Building modules, stage 2.
MODPOST 1 modules
make[1]: 离开目录"/usr/src/linux-headers-5.4.0-150-generic"
lsmod wch.ko || true
mkdir -p /lib/modules/5.4.0-150-generic/kernel/drivers/tty/serial || true
cp -f ./wch.ko /lib/modules/5.4.0-150-generic/kernel/drivers/tty/serial || true
depmod -a
ub18@ub18:~/Driver/Linux$
```

在Linux文件夹路径下，键入ls /dev命令，待编译后，可看到安装成功的串口 ttyWCH0, ttyWCH1, ttyWCH2, ttyWCH3, ttyWCH4, ttyWCH5, ttyWCH6, ttyWCH7。

ttyS11	ttyS26	ttyWCH3	vcs5	vflo
ttyS12	ttyS27	ttyWCH4	vcs6	vga_arbiter
ttyS13	ttyS28	ttyWCH5	vcsa	vhci
ttyS14	ttyS29	ttyWCH6	vcsa1	vhost-net
ttyS15	ttyS3	ttyWCH7	vcsa2	vhost-vsock
ttyS16	ttyS30	ubuntu-vg	vcsa3	zero
ttyS17	ttyS31	udmabuf	vcsa4	zfs
ttyS18	ttyS4	uhid	vcsa5	
ttyS19	ttyS5	uinput	vcsa6	
ttyS2	ttyS6	urandom	vcsu	
ttyS20	ttyS7	userio	vcsu1	
ttyS21	ttyS8	vcs	vcsu2	
ttyS22	ttyS9	vcs1	vcsu3	
ttyS23	ttyWCH0	vcs2	vcsu4	
ttyS24	ttyWCH1	vcs3	vcsu5	
ttyS25	ttyWCH2	vcs4	vcsu6	

驱动软件下载以及详细规格和说明书请扫二维码

