# µPAC-7186E 系列

#### µPAC-7186E 系列软件用户手册



#### 承诺

郑重承诺: 凡泓格科技股份有限公司产品从购买即日起一年 内无任何材料性缺损。

#### 免责声明

凡使用本系列产品除产品质量所造成的损害, 泓格科技股份 有限公司不承担任何法律责任。 泓格科技股份有限公司有义 务提供本系列产品可靠而详尽资料, 但保留修订权利, 且不 承担使用者非法利用资料对第三方所造成侵害构成的法律责 任。

#### 版权

版权所有 © 1999-2007 泓格科技股份有限公司,保留所有权力。

#### 商标

手册中所涉及所有公司商标,商标名称及产品名称分别属于 该商标或名称的拥有者所有。

1. 绪论	1
1.1 对照表	2
1.2 三类典型应用	4
1.2.1 虚拟串口应用	5
1.2.2 以太网 I/O 应用	6
1.3 软件文件树	8
1.4 软件安装	9
1.4.1 安装MiniOS7 Utility	9
1.4.2 PC诊断工具	10
2. 7186E系列MINIOS7	12
2.1 мPAC-7186E系列操作系统MiniOS7	12
2.2 мРАС-7186E系列范例程序	14
3. VXCOMM应用	17
2.4 瓶注	17
3.1	10
3.2.1 纽构	10 18
3.2.2 圳口顷剂	10
3.3 添加MPAC-7186F服条并配置VxCOMM 驱动	13
3 4 移除MPAC-7186F服务	28
3.5 诊断及故障排查	30
3.5.1 诊断	30
3.5.2 常见问题解答	37
4. 以太网 I/O 应用	41
	11
4.1 ASERVER朱构原理	4 I 12
4.2 A SERVER 单文仍以	<del>4</del> 2 42
4.2.7 1	42
<b>423</b> 命令行操作	12
4.2.4 Xserver构架流程图	56
4.3 XSERVER构架例程清单	59
4.4 基于XSERVER构架程序设计	63
4.4.1 Xserver初始源码	63
4.4.2 推荐使用BC++ 3.1 编译程序	66
4.5 基于XSERVER构架例程	74
5.9术语表	87

# 1 绪论

μPAC-7186E 系列控制器配备有 RJ-45 接口,一个 I/O 扩展总线及数个 串口。μPAC-7186E 作为嵌入式控制器,使用 C 语言进行编程,并提供 完整的库文件及相关的 Demo 例程以供使用,可完美控制透过以太网或 串口 RS-232/485/422 之设备。此外,每款 μPAC-7186E 控制器拥有一 个 I/O 扩展总线可方便快捷使用诸如: D/I、D/O、A/D、D/A、定时器/频 率计、Flash 扩展记忆体及配有后背电池的 SRAM 等功能各异的扩展 板。



### 1.1 对照表

µPAC-7186E 与 I-7188E 硬件功能对照表:

	μPAC-7186E	I-7188E
CPU	80186, 80MHz	80188, 40MHz
SRAM	512K Bytes	512K Bytes
Flash Memory	512K Bytes	512K Bytes
EEPROM	16K Bytes	2K Bytes
NVRAM	31 Bytes (后背电池,数据 10 年有效)	31 Bytes (后背电池,数据 10 年有效)
RTC (实时时钟)	Yes	Yes
64-bit 硬件序列号	Yes	Yes
内置看门狗时钟	Yes	Yes
СОМ1	RS-232 (TXD, RXD, RTS, CTS, GND)	RS-232 (TXD, RXD, RTS, CTS, GND)
COM2	RS-485 (D2+, D2-)	RS-485 (D2+, D2-)
Ethernet 接口	10/100 Base-TX	10 BASE-T
I/O 扩展总线	Yes	Yes
用户自定义 I/O 引脚	14 Pins	14 Pins
保护	电源极性反向保护	电源极性反向保护
壳体接地,ESD 保护	Yes	No
电源供电	+10 to +30VDC (非稳压)	+10 to +30VDC (非稳压)
功耗	<b>1.5W</b> : μPAC-7186EX <b>2.5W</b> : μPAC-7186EXD	2.0W: I-7188EX 3.0W: I-7188EXD
操作系统	MiniOS7	MiniOS7

	μΡΑC-7186Ε		I-7188E	
	函数库名称	头文件名称	函数库名	头文件名称
模块函数库	7186el.lib	7186.h	7188el.lib	7188.h
TCP/IP 函数库	TCP_dm32.Lib	TCPIP32.h	TCPIP32.Lib	TCPIP32.h
Vxcomm 函数库	vcomnnnn.Lib	Vxcomm.h	vcomnnnn.Lib	Vxcomm.h
Xboard 函数库	X107.lib X109.lib	X107.h X109.h	X107.lib X109.lib	X107.h X109.h
	更多	更多	更多	更多

µPAC-7186E 与 I-7188E 软件功能对照表:

**注意:** Vxcomm 库文件名称模式为: vcomnnnn.exe nnnn: 代表 Vxcomm 库文件版本号。

全部函数库文件均可从以下方式获得:

CD:\Napdos\7186e\tcp\xserver\demo\lib

CD:\Napdos\7186e\demo\lib\xboard

http://ftp.icpdas.com/pub/cd/8000cd/napdos/7186e/tcp/xserver/demo/lib http://ftp.icpdas.com/pub/cd/8000cd/napdos/7186e/demo/lib/xboard

使用以上函数库,注意事项如下:

- 版本号为 3225 或更新的 vxcomm 库文件均适用于 μPAC-7186E 和 I-7188E 系列。
- 使用 I-7188 系列编程,采用版本号为 3225 或更新的 vxcomm 库文件 时,与之相对应的 TCP/IP 函数库必须使用 2007 年 7 月 22 日或更新 版本。
- 3. 使用库文件 TCP\_dm32.Lib 或 TCPIP32.Lib 时,可使用头文件 TCPIP32.h.
- 4. 除 vcom3225.lib 之外的其它函数库, 仅适用于特定模块。

所有有关 I-7188E 系列例程均适用于 μPAC-7186E。注意,在使用 μPAC-7186E 例程编译前,请更换相应的函数库以适用于 μPAC-7186E。

### 1.2 三类典型应用

μPAC-7186E 系列控制器通常有三类典型应用(虚拟串口技术,以太网 I/O 和 Web 服务应用),而这些应用则使用着不同固件和程序类型。用 户可根据系统应用选择相应的类型。

使用虚拟串口应用,µPAC-7186E 作为 RS-232 到以太网的转换器,可使 PC 机控制 256 个串口(包括真实串口)。

而使用以太网 I/O 和 Web 服务应用,用户也可使用 µPAC-7186E 程序控制(基于 Xserver 和 Web Server 构架)。



### 1.2.1 虚拟串口应用



虚拟串口应用使用方法如下:首先,安装虚拟串口驱动程序;安装后,使用 VxComm Utility 可映射远程 μPAC-7186E 串口作为 PC 机本机串口,使用方式与本机现实串口(COM1)一致,可直接控制串口上其它设备。这样,使用 μPAC-7186E 的虚拟串口技术,最多可使一台 PC 机虚拟 256 个串口。

另外,用户不必分心在网络程序编写,VxComm 驱动可自动处理以太网信息。

#### 虚拟串口 VxComm 优势:

- 1. 用户可透过以太网方便地进行用户升级,而无须修改代码。
- μPAC-7186E 可支持多客户端,每台 μPAC-7186E 可同时处理 30-N 个 socket 连接,其中 N 为 μPAC-7186E 串口数。

例:如果主机 PC 使用了 µPAC-7186E 上两个虚拟串口,那么 µPAC-7186E 最多允许 14 台 PC 相接。

### 1.2.2 以太网 I/O 应用

![](_page_7_Figure_1.jpeg)

μPAC-7186E 内建灵活的 Xserver 架构作为以太网 I/O 应用设计程序框架,用户可通过修改 Xserver 控制所有 μPAC-7186E 硬件:串口,I/O 扩展板,5位7段 LED 和其它相关产品。

### Xserver 构架优势:

- 1. 我们为所有客户设计,修改,升级最新 Xserver 架构。
- 2. 用户可通过开发工具(如: TC/BS/MSC )方便的修改 Xserver 架 构,所有相关库文件均以C语言编写。
- 3. 高速运行速度。Xserver 程序(Demo4.exe)可以以每秒 1360 个扫描循 环运行。
- 4. Xserver 架构诸如: 以太网通讯,程序循环等大多数代码已经在 VxComm.lib 中完成,大大降低了用户开发周期。

- 5. 对于 Xserver 架构,用户仅只需修改 6 个功能子函数。
- 6. 已设计好的命令协议使用于大多数 µPAC-7186E 的需求。
- 7. 用户也可方便地开发扩充自定义协议。
- 8. 支持多客户端。Xserver 构架可同时处理 30-N 个 socket 连接,其中 N 为 μPAC-7186E 串口数。
- 9. 拥有自动唤醒机制。Xserver 构架可检测封包是否超时,一旦软件发生崩溃,Xserver 将自动重启。
- **10**. 完全免费提供基于 Xserver 的例程(TC/BC/MSC)及相关客户端 Demo 程序(VB/VC)。

### 1.3 软件文件树

为帮助用户降低开发时间,我们提供各式软件资源以共分享,其中包括 有:说明文档、驱动、库文件、诊断程序、客户端程序及大量 Xserver 例程源代码。用户可通过下面的文件树快速找到自己所需要资源。

### μPAC-7186E 子文件树:

![](_page_9_Figure_3.jpeg)

**注意:**软件及用户手册将会实时更新,因此随机赠送的 CD 中也将不定时升级。故使用前最好读取文件本地目录中 README.TXT 文件,以获得文档最新资讯。

### 1.4 软件安装

### 1.4.1 安装 MiniOS7 Utility

步骤 1: 请以如下方式获得安装软件:

- CD:\8000\Napdos\MiniOS7\utility\MiniOS7\_utility\
- http://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/utility /minios7\_utility/

#### 步骤 2: 进入文件下载目录,在 Windows 中双击安装文件并执行。

![](_page_10_Picture_6.jpeg)

步骤 3: 软件 MiniOS7 Utility 安装完成后,单击 Finish 按钮退出安装。

![](_page_10_Picture_8.jpeg)

### 1.4.2 PC 诊断工具

MiniOS7 Utility 提供 PC 串口诊断工具。该 PC 诊断工具可从 MiniOS7 Utility 上端工具下拉菜单中获得。

🚵 Mini0	0S7 Utility Ve	rion 3.1.1 (bu	ild 3.1.1.1)		
🔉 File	Connection	🝷 🚸 Command	😨 Configuration	🛅 Tools 🥔 Help 🔻	
				7188XW 7188EU 7188E	
				Send232 SendTCP VxComm Utility	
				Console F10	

PC 诊断工具包括有:

● Send232: 用户使用串口(RS-232) 与设备连接时,可测试用于虚拟串口技术。

🚰 Send232 V. 2.0.1 COM1	
COM status COM1 9600 Line control : N.8,1 Open Close	Send string with None CLF_CR CR CR_LF CLF
Auto send Interval 500 Set Send Stop	String Send
Send	Receive
Clear	Clear Exit Program

详情请参考章节 3.5 更多操作细节

SendTCP: 使用 TCP 协议与 µPAC-7186E 上串口设备进行通讯调

😻 SendTCP:7188E/8000E Diagnostics App. [v2.08.02, Apr.2	7, 2004]
IP and TCP Command port       Send Command To 7188E         IS2.168.255.1       10000         Connect       Disconnect         Select Port:       Send Data with         Image: Select Port:       Select Port:         Image: Select Port:       Sel	Network Status   Gateway   Mask:   N/A   Set   MAC:   N/A   7188E COM Status   N/A   Set   Fw. Ver.   N/A   MiniDs7 Version   N/A   Close

● 7188E: 使用命令提示符,以 TCP 协议发送数据到指定设备。

方法如下: 7188e [-S:IP] [-P:Port]: TCP 协议连接设备。

\*Q:退出程序并断开连接

C:\Program Connect to	Files\7188E\PCDiag>7188e 192.168.30.24:10000	-s:192.168.30.24	-p:10000
01 v3.0.01[11∕ 10	19/2001 ]		
7188E2 *q			
C:\Program	Files\7188E\PCDiag>		

# 2 7186E 系列 MiniOS7

### 2.1 µPAC-7186E 系列操作系统 MiniOS7

MiniOS7 为泓格科技股份有限公司自主设计的类 DOS 嵌入式操作系统, 拥有自主知识产权。在大多数条件下, DOS(无论是 PC-DOS、 MS\_DOS 或 ROMDOS)可完成一系统命令代码,以告知计算机如何处理 信息。而 DOS 可运行程序,管理文件,控制信息处理,输入输出及执行 大量其它相关功能。

MiniOS7 可支持如下系列:

- 7188XA/7188XB/7188XC 系列
- 7521/7522/7523 系列
- 7188EA/7188EX 系列
- 7188E1/7188E2/7188E3/7188E4/7188E5/7188E8 系列
- µPAC-7186E 系列
- 8000 系列
- iView-100 系列
- 更多新型嵌入式控制器

### 如下库文件适用于 µPAC-7186E:

- 7186el.lib: CPU 及 I/O 相关库文件(Large model)
- TCP\_dm32.lib: TCP/IP 相关库文件(Large model)
- VcomNNNN.lib: Xserver 相关库文件(Large model), 其中 NNNN 为库文件版本号

函数功能	范例函数	
	InstallCom1, InstallCom2,	
电口堀佐	IsCom1, IsCom2,	
中口珠作	ToCom1, ToCom2,	
	ReadCom1, ReadCom2,	
EEPROM	WriteEEP, ReadEEP, EnableEEP, ProtectEEP	
NVRAM 及 RTC	ReadNVRAM, WriteNVRAM, GetTime, SetTime, GetDate, SetDate	
IED及后位IED目示	LedOn, LedOff, LedToggle, Init5DgitLed, Show5DigitLed,	
LED 及 5 位 LED 亚小	Show5DigitLedWithDot	
Flash 记忆体	FlashReadId, FlashErase, FlashRead, FlashWrite	

库文件 7186el.lib 支持如下函数:

	TimerOpen, TimerClose, TimerResetValue, TimerReadValue
时钟及看门狗	StopWatchReset, StopWatchRead, StopWatchStop
	InstallUserTimer EnableWDT, DisableWDT, RefreshWDT
文件揭作	GeFileNo, GetFileName, GetFilePositionByNo,
义什 <sup>採</sup> IF	GetFilePositionByName
连接 I-7000 系列模块	SendCmdTo7000, ReceiveResponseFrom7000
可编程 I/O	SetDio4Dir, SetDio4High, SetDio4Low, GetDio4
其它	Kbhit, Getch, Putch, LineInput, Scanf

详情请参考如下文档:

- CD:\Napdos\7188e\document\tcpiplib.pdf
- CD:\Napdos\MiniOS7\document\Lib\_Manual\_For\_7188E\index.htm

## 2.2 μPAC-7186E 系列范例程序

μPAC-7186E 设计有大量范例程序,所有程序源代码均可以随机附赠的 光盘中获得。用户可在初次使用时,通过修改编译特定例程大大提高上 手速度。所有例程功能参考如下:

文件	<b>Demo</b> 例程	说明
	Hello_C	检测操作系统是否为 MiniOS7.
Hello		<b>注意:</b> MSC 并不支持 C++语言。 Hello_C++文件仅支
		持 BC.
		1. 如何完成输入数据函数
	C_Style_IO	2. 如何获得字符
		3. 如何使用 C 函数: sscanf, 或仅使用 Scanf().
COM_Port	Receive	从串口接收数据
		PC 给 7186E 发送命令,并从 7186E 接收响应。如何
		使用串口及 LED 以显示相关信息便于调试程序。
	ToCom_In_Out	通过串口读/写数据
DeteTime		从实时时钟读取数据及时间并显示与显示器上(用户亦
Daterime		可设定数据及时间).
	Led	如何使用 DelayMs 函数调节 LED 状态 ON 或 OFF
	Seg7led	控制红色 LED 及 5 位 7 段 LED.
		在许多应用中,针对一些关键数据常常存储在文本里,
	Config_1_Basic	因而程序往往需要读取其中数据. FSeek 可用于检索
Filo		文本中关键信息。
LIIG		除去 config_1_Basic 之外,同时还加入了函数
	Config_2_Advanced	GetProFileInt, GetProFileFloat 及 GetProFileStr.。
		这些函数可确定文本文件类型。
	Demo5	说明如何使用 NVRAM.
	EEPROM	向 EEPROM 写入数据并显示到屏幕上。
	EEPROM-r	读取已经写入 EEPROM 中数据
	EEPROM-w	输入一值,并存储到 EEPROM 中 block 1 每一地址中
		(数值将以1累加).
Memory	Flash	读,写和清除 Flash 记忆体
	Flash-r	读取已经写入 Flash 记忆体中数据
	Flash-w	输入数据并写入 Flash 记忆体中(数值将以 1 累加).
	NVRAM-r	从 NVRAM 读取已写入数据
	NVRAM-w	向 NVRAM 中写入数据 (数值将以 1 累加).
	Top-Mem	AllocateTopMemory 函数使用例程
Misc	Reset	复位初始值
	Runprog	使用 Ungetch 函数运行另外一个程序
	SerialNumber	检索 I-7188XA(D)产品序列号
	Watchdog	激活 Watchdog 或绕过激活 Watchdog.
7K87K_Module	7K87K_demo_for_com	如何通过 COM2 连接 7K 及 87K 系列模块
	7K87K_AI_for_Com	
	7K87K_DI_for_Com	

	7K87K_DIO_for_Com	
	7k87K_DO_for_Com	
	AO_024_for_Com	
	AO_22_26_for_Com	
	Demo90	Timer 函数使用例程.
	Domo01	利用 CountDownTimer 函数调节 LED 状态 ON 或
	Demogi	OFF.
Demo92	Demo92	利用 StopWatch 函数调节 LED 状态 ON 或 OFF.
Timer	Demo96	使用 InstallUserTimer 函数控制 5 位 7 段 LED.
	Demo97	使用 DelayMs 函数调节 LED 状态 ON 或 OFF.
	Deme08	如何利用 timer 函数通过 I-7188XA(D)发送/接收 7000
	Demoag	系列模块数据.
VD a and		该项目包含所有 I/O 应用于 the I-7188XA(D).扩展板例
XBOald		程

**注意:µPAC-7186E** 系列与 **7188E** 系列在功能和使用方式上极为相近,因此所有有关 µPAC-7186E 系列例程请参考 **7188E** 系列例程说明。

所有相关 Demo 例程均可以如下方式获得:

- http://ftp.icpdas.com/pub/cd/8000cd/napdos/7188e/demo/
- CD:\Napdos\7188e\demo\

使用 μPAC-7186E 系列编译 Demo 例程前,请先行将旧有的库文件 7188el.lib 更换为 7186el.lib。

所有函数库文件可从如下方式获得:

- http://ftp.icpdas.com/pub/cd/8000cd/napdos/7186e/demo/lib/
- CD:\Napdos\7186e\demo\lib\

部分典型 TCP/IP 例程如下:

典型 TCP/IP 例程	μPAC-7186E	PC
Ping demo	Client, ping.exe	None
Telnet server demo	Server, telserv.exe	Client, telnet.exe
Telnet server demo2	Server, telserv2.exe	Client, telnet.exe
Demo1: TCP/IP demo	Server, demo1.exe	Client, Client1.exe
Demo2: TCP/IP demo	Server, demo2.exe	Client, Client1.exe
Demo3: TCP/IP demo	Server, demo3.exe	Client, Client1.exe

**注意:µPAC-7186E** 系列与 **7188E** 系列在功能和使用方式上极为相近,因此所有有关 µPAC-7186E 系列例程请参考 **7188E** 系列例程说明。

所有相关 Demo 例程均可以如下方式获得:

- http://ftp.icpdas.com/pub/cd/8000cd/napdos/7188e/tcp/demo/
- CD:\Napdos\7188e\TCP\demo\

使用 μPAC-7186E 系列编译 Demo 例程前,请先更换库文件(诸如: 7186el.lib, TCP\_dm32.lib 和 VcomNNNN.lib)以适用于 μPAC-7186E。

所有函数库文件可从如下方式获得:

- http://ftp.icpdas.com/pub/cd/8000cd/napdos/7186e/tcp/xserver/de mo/lib/
- CD:\Napdos\7186e\TCP\xserver\demo\lib\

更多有关 Xserver 构架的 TCP/IP 例程请参考章节 4.3。

# 3. VxComm 应用

- 概述
- 安装 VxComm 驱动
- 添加 µPAC-7186E 服务并配置相应串口
- 移除 µPAC-7186E 服务
- 诊断及故障检修

### 3.1 概述

安装配置 VxComm (虚拟串口) 驱动及 VxComm Utility 易于安装及使用,本文档将详细阐明如何正确安装。 快速上手手册共分为三个部分:

- 一、如何安装软件
- 二、如何添加 µPAC-7186E 服务并配置相应串口
- 三、如何移除 µPAC-7186E 服务。

### 3.2.1 结构

透过 VxComm 驱动创建串口,并将远程 µPAC-7186E 串口映射于 PC 机。用户仅仅只需通过 RS-232 客户端调节设置到基于以太网映射 µPAC-7186E 的串口上即可正常使用并可与相应的串口设备通讯。

![](_page_19_Figure_2.jpeg)

### 3.2.2 端口映射

Vxcomm Driver/Utility 支持与 µPAC-7186E 对应 COM1 到 COM8 的端口 Port 1 到 Port 8。其它 I/O 端口用于安装于 µPAC-7186E 上的 I/O 底板。

在 VxComm Driver/Utility 支持下,用户可方便的映射远程串口及 I/O 底 板为本机的虚拟串口。每台 PC 上可最多控制 256 个串口(包括本机自带 串口 COM1 和 COM2).

本地串口 (主机 PC)	VxComm Driver/Utility (主机 PC)	远程串口 (µPAC-7186E)
COM ?	Port 1	COM1
COM ?	Port 2	COM2
COM ?	Port 3	COM3
COM ?	Port 4	COM4
COM ?	Port 5	COM5
COM ?	Port 6	COM6
COM ?	Port 7	COM7
COM ?	Port 8	COM8
COM ?	Port I/O	I/O 扩展板

### 3.2 安装 VxComm 驱动

步骤 1: VxComm 安装软件可从如下方式获得:

- http://ftp.icpdas.com/pub/cd/8000cd/napdos/7188e/tcp/vxcom m/driver(pc)/
- CD:\8000\Napdos\7188e\TCP\vxcomm\driver(pc)\

根据 Windows 操作系统选择合适版本:

- vxcomm2K\_v20811.exe for Windows 2000/XP
- vxcommt\_v20811.exe for Windows NT
- vxcomm98.exe for Windows 98

步骤 2: 进入软件安装文件夹,双击文件图标并执行程序。

🕰 VxComm Driver Setup	
VxComm Driver v2.08.12 Beta 3 for Windows 2000/XP/2003	
Supports 71 Vacamenta Setun	
Choose Destination Location Select folder where setup will instal files.	
Setup will install VxComm2K in the following folder. To install to this folder, click. Next. To install to a different folder, click. Browse and select another folder.	
Destination Folder C.VLCPDASWsComm_Driver_2K Bjowse	
InstallShield Cancel	
🛃 start 🔰 🌉 VxComm Driver Setup 📁 Liam	😯 🏴 🦁 5:15 PM

步骤 3: 选择"Yes, ..." 并单击 Finish 按键重启计算机。

VxComm Driver Setup	
VxComm Driver Setup	InstallShield Wizard Complete The InstallShield Wizard has successfully installed VxComm2K. Before you can use the program, you must restart your computer. P Yes, I want to restart my computer now. 个 Yes, I want to restart my computer now. No, I will restart my computer later. Remove any disks from their drives, and then click Finish to complete setup.
InstallShield	步骤 2 <back cancel<="" finish="" td=""></back>

步骤 5: 计算机重启后, VxComm Utility 将会自动询问虚拟串口配置。 更多信息请参考下一章节。

### 3.3 添加 µPAC-7186E 服务并配置 VxComm 驱动

步骤 1: 确定 µPAC-7186E 正在运行 VxcomNNNN.exe 。文件 VxcomNNNN.exe 可从如下方式获得:

- http://ftp.icpdas.com/pub/cd/8000cd/napdos/7186e/tcp/vxcom m/server(7186e)/
- CD:\8000\Napdos\7186e\TCP\vxcomm\server(7186e)\

**注意:**版本格式名称为: VxcomNNNN.exe NNNN:版本序列号

- 步骤 2: 获得 µPAC-7186E 的 IP 地址。 工具 MiniOS7 Utility 透过串口进入 µPAC-7186E 操作系统可有效获得 IP 地址。
- 注意: µPAC-7186E 默认 IP 地址为 192.168.255.1.
- 步骤 3: 进入 Winsows 开始菜单, 依次进入"程序/7188e/Vxcomm2K/ 或本地 VxComm Utility".

![](_page_22_Picture_8.jpeg)

步骤 4: 在使用 µPAC-7186E 前, 确定 IP 正确有效。输入 µPAC-7186E 的 IP 地址, 单击按钮 Add Server 添加虚拟串口服务。

🐗 VxComm Utility [ v2.8.1	2 Beta, Jun, 27, 2007]	
File Server Port Tools		
步骤	1 E/8000E Internet/Ethernet Controller	
ICP 53	[IP: 192.168.255.1] ort: 10000 Tir	neout (ms) 5000 🔊 Add Server
DAS	Check Duplicated IP	enver 牛爾 2
Configure Server		Port COM Baudrate
X Remove Server		
Configure Port		
Web Configuration		
System Infomation		
Search Device	Name Alias MAC Address IP Address S	Gub-net Gateway DHCP
Configure Device		
Exit		
7186EX:10000, v3.2.25[06/21/200	)7], found!!	

### 注意:

● 核实 IP 项目

该项目将默认检测自动检测 IP 地址是否已在服务器窗口(左边窗口) 中存在。若 IP 地址已经在服务中,即将提示报警信息。

![](_page_23_Picture_4.jpeg)

● 连接服务器项目

该项目将默认自动连接 µPAC-7186E 并检索到设备名称,添加到服务器客窗口中 (左边窗口)。若单击 Add Server 前没有检测 Connect to Server,将会出现如下报警窗口。选择正确 µPAC-7186E 模块序号并单击按键 OK。

Dialog	X
Selecting correct model number	ОК
Model Number: 7186EX	Cancel

● 超时 (ms) 设置:

该超时值用于服务 VxComm Utility 连接、断开服务;发送、接收数据,检测是否超时。

● 命令端口设置:

初始情况下,µPAC-7186E 命令配置的 TCP 端口号为 10000。若 µPAC-7186E 设置已经改变,则必须指定正确地址以使 VxComm Utility 及驱动可连接到设备。

步骤 5: 选取 µPAC-7186E 模块并双击 Port1, Port2 等设置虚拟串口。

L					
'	V×Comm Servers	Port	COM	Baudrate	
L	7186EX (192.168.255.1)	Port I/O	UnMap	Fixed	
'		Port 1	UnMap	Dynamic	
I		Port 2	UnMap	Dynamic	
1					
I					
I					
I					

步骤 6: 选取适当的串口号,并单击按键 OK。

Port Configuration		×
Server: 7186EX	(10.0.8.136), Port 1	
Port Mapping (F	PC) Port Setting (Device)	- 1
Select COM	COM10 V	
Re-assign Skip baud Virtual CO (Less continue)	COM10 COM11 COM12 COM12 COM13 COM13 COM13 COM14 COM14 COM15 COM15 COM16 COM16 COM17 COM18 COM19 COM20 COM21 COM23	

注意:

- Re-assign COM number for all subsequent ports option 该选项将自动分配未使用的串口号顺序给各端口。
- Use 7188E/8000E current setting (Fixed Configuration) 若使用该端口配置功能,该端口将不再可以动态调整串口波特率及 数据格式。
- 步骤 7: 选取其中一个 µPAC-7186E,并单击按钮 Web Configuration 进入设置选项。

Server Configuration         Configure Server         Remove Server         Configure         Configure         Web Configuration         System Infomation         IP Address :	VxComm Utility [ v2.8.12 File Server Port Tools	Beta, Jun. 27, 2007]         7188E/8000E Internet/Ethernet Controller         IP:       192.168.255.1         Port:       10000         IV:       Check Dunlicated IP         IV:       Connect to
	Configure Server Configure Web Configuration System Infomation	Server Configuration         Server : 7186EX [10.0.8.136]         Server Options [PC]       Device Information         The following items are all PC side settings, not device settings.         Keep Alive Time (Seconds) :       120         commend : 7188E/8KE's System Timeout * 1/3         Connect Timeout (Seconds) :       180         Connect Timeout (Seconds) :       5         Command Port (TCP):       10000         Virtual I/O Port (TCP):       9999         IP Address :       10.0.8.136

步骤 8: 输入设置。新设置将在 Server Options 窗口中显示出来,单击 OK 保护最新设置。

注意:

### ● Keep Alive Time (ms) field (持续激活功能):

μPAC-7186E 正常接通后,VxComm 驱动将自动周期性发送指使 保证 μPAC-7186E 持续在激活状态,且指令成功发出后时钟将会复 位。而持续激活机制将在下个周期内激活。

持续激活周期默认值为 7000 ms。推荐设置为 μPAC-7186E's 系统 超时时长的三分之一或更短。

#### ● Connection-Broken (ms) field (断线检测功能):

若 μPAC-7186E 与系统断线后,VxComm 驱动将尝试自动连接。 此时,客户端向 μPAC-7186E 发送指令,若在以太网层若在指令发 送后 20 秒或更多时间发送失败,则会响应"Disconnect"事件给 VxComm 驱动。用户可自定义断线检测时钟 Connection-Broken time (例: 10000 ms),强行使 VxComm 驱动以更快的时间响应并 自动重新连接。

若在断线检测前 Connection-Broken time,系统没有送出/接收到 任何信号,将会认定为断线, VxComm 驱动也将再次重启。因 此,在连机时,确保持续激活周期 Keep-Alive Time 应小于断线检 测时钟 Connection-Broken time。

默认 μPAC-7186E 系统超时 System Timeout (/STxxx)值 为 300 秒。当客户端程序连接到 μPAC-7186E,其将会持续发送指使以保 持 μPAC-7186E 在超时前激活状态,否则 μPAC-7186E 将会自动 重启,且客户端程序必须再次重新连接 μPAC-7186E。

若用户将 Keep-Alive Time 及 Connection-Broken time 在该机制下设置为 0 或禁用,则系统超时也将必须设置为 0 或禁用复位机制。

● Connect Timeout (ms) field (连接超时功能): 当连接或断开时,其超时值将被写入 MS TCP/IP 驱动以作参考。

#### ● Command TCP Port field(TCP 端口指令):

默认设置下,µPAC-7186E 使用 TCP 端口 10000 为命令/ 配置端口。若改变 µPAC-7186E 端口,则必须重新指派正确端口,这样 VxComm 驱动 才可连接正确 TCP 端口。

该 TCP 端口可用于设置波特率,数据格式,CTS/RTS 控制模式等。

#### ● Port7000 Port field(7000 端口):

默认设置下,µPAC-7186E 使用 TCP 端口 9999 为 Port7000,该 TCP 端口将被保留。 步骤 9: 单击 Exit, 在退出 VxComm Utility 前 Re-starting the Driver 对话框将自动弹出,再点击按钮 Re-start 重启驱动。

Web Configuration	
System Infomation	
	VxComm Utility : Re-starting the Driver
Search Device	To use the new configuration, please close all virtual COM ports first, and then
Configure Device	click the "Re-start" button to stop/start the driver.
步骤1	Status: Driver is running. 步骤 2 Re-start Cancel
Exit	
7186EX:10000, v3.2.25[06/21/2007	], found!!

### 3.4 移除 µPAC-7186E 服务

### 步骤 1: 打开 VxComm Utility。

步骤 2: 选择将要移除服务器,并单击按钮 Remove Server。

🛷 VxComm Utility [ v2.8.1	2 Beta, Jun. 27, 2007]
File Server Port Tools	
<b>CREEK</b>	7188E/8000E Internet/Ethernet Controller         IP:       192.168.255.1         Port:       10000         Timeout (ms)       5000         IP:       Check Duplicated IP         IP:       Connect to Server
Configure Server	
Remove Server Configure Port	步骤 2
Web Configuration	
System Infomation	
Search Device	Name Alias MAC Address IP Address Sub-net Gateway DHCP
Configure Device	
Exit	
	] ]]] found!!
7100EA:10000, V3.2.23[06/21/200	y () roana:

步骤 3: 在弹出确认窗口后, 核定选择并单击按钮 Yes 移除。

VxComm Utility		
Server : 7186EX (	192.168.255.1)	
Are you sure to remove this server?		
Yes	No	

V 泳 4. 千山以山 LAIL A	区山(主/]、。	
🛷 VxComm Utility [ v2.8.12 Beta, Jun	.27, 2007]	
File Server Port Tools		
7188E/80           IP:           IV:	200E Internet/Ethernet Controller 2.168.255.1 Port : 10000 Timeout (ms) 5000 Check Duplicated IP I Connect to Server	Add Server
Configure Server	m Servers Port COM Baudr	ate
Remove Server		
Configure Port		
Web Configuration		
System Infomation		
Search Device Name #	Alias MAC Address IP Address Sub-net Gateway DHC	P
Configure Device		
Exit		
Status		

### 3.5 诊断及故障排查

### 3.5.1 诊断

通过使用工具 VxComm Utility 配置完 VxComm 驱动后, VxComm 驱动即可正常工作。此外,用户也可使用简单的测试程序确定是否正常工作。

注意: 部分测试工作将使用其它设备及客户端程序。

#### ■ 例 1: Loop-Back Testing (环路信号回送检测)

步骤 1: 确定 VxComm 服务工作在/m0 模式下, 批处理文件

Autoexec.bat 应包含如下内容:

🖪 Untitled - Notepad		×
File Edit Format View Help		
runexe /2 /ST300 /mO		<
		v
8	>	:

注意:详情请参考章节4.2.3 "命令行项目"。

步骤 2: 连接 µPAC-7186E 引脚 TXD1 到 RXD1。

![](_page_31_Figure_10.jpeg)

步骤 3: 使用 VxComm Utility 设置 µPAC-7186E 的 COM1 作为主机 PC 的 COM4。

# 步骤 4: 在 MiniOS7 Utility 项目 Tools 下拉菜单栏中选择项目 Send232, 单击并进入串口应用程序。

MiniOS7 Utility Verion 3.1.1 (build 3	.1.1.1)	
🔯 File 👂 Connection 👻 🚸 Command 🗾 🤇	Configuration	Tools 🐠 Help 🔻
Look jn: MiniOS7_Utility	• 0	7188XW 7188EU 2188E
Name Size	Туре	No Name
bin     FIRMWARE     OS_IMAGE     icpdas     1KB	File Folder File Folder File Folder Internet	Send232 SendTCP VxComm Utility
Ioad232.dll     88KB       MiniOS7_Utility     1,025KB       MiniOS7_Utility     2,094KB       MiniOS7_Utility     3KB       Ioad232.dll     3KB       Ioad232.dllllll     3KB       Ioad232.d	Applicat Compile Applicat Configu Applicat DAT File Applicat	COM status         COM1 ▼ 9600 ▼         Line control:       N.8.1         Open       Close
	13	Auto send Interval 500 Set Send
		Send Stop Send Receive
<		Exit Program

步骤 5: 选择 COM4,并单击按钮 Open 开启主机 PC 上已虚拟的 COM4。

![](_page_32_Picture_3.jpeg)

步骤 6: 可以 Send 栏中输入任意字符并点击下方的 Send 按钮发送。 字符将从主机虚拟的串口 COM4,也就是 µPAC-7186E 的 COM1发送(如上图路径 Path1),此时主机 PC 虚拟的 COM4 口 将立刻接收到 µPAC-7186E 上串口 COM1 上的数据(如上图路径

Path2),并显示在应用程序接收栏中。						
	s <sup>all</sup> Se	🕬 Send232 V. 2.0.1 COM4				3
		OM status	00 V.8,1 N.8,1 Close Set	Send strin	g with O LF_CR O CR_LF 步骤 1	步骤 2
	Se Vir	nd tual COM		Virtual COM	步骤 3	
		Clear		C Exit F	lear <sup>D</sup> rogram	

- 例 2: Close-Loop Testing (闭环测试)
- 步骤 1: 确定 VxComm 服务工作在/m0 模式下,批处理文件 Autoexec.bat 应包含如下内容:

📕 Untitled - Notepad	$\mathbf{ imes}$
File Edit Format View Help	
runexe /2 /ST300 /mO	<
	$\sim$

步骤 2: 确定连接方式如下图:

![](_page_34_Figure_0.jpeg)

步骤 3: 在 MiniOS7 Utility 项目 Tools 下拉菜单栏中选择项目 Send232, 单击并进入串口应用程序。

步骤 4: 选择 COM1,并单击按钮 Open 开启主机串口 COM1。

![](_page_34_Picture_3.jpeg)

步骤 5: 使用 VxComm Utility 设置 µPAC-7186E 串口作为主机 PC 虚拟 串口 COM1。

步骤 6: 执行另一个 Send232 应用程序,开启主机虚拟的 COM4。

![](_page_34_Picture_6.jpeg)

步骤 7: 在先开启的程序中输入"COM1",并单击按钮 Send 数据将由主机 PC 的 COM1 经由路径 Path1 发送,此时 µPAC-7186E 的串口接收到数据将立即通过路径 Path2 返回到主机虚拟的 COM4中。

步骤 8: 在后开启的程序中输入 "Virtual COM", 并单击按钮 Send 数据将 通过路径 Path2 发送到 µPAC-7186E,此时串口接收到数据后 将立刻通讯路径 Path1 返回到主机 PC 的串口 COM1。

#### ■ 例 3: External-Devices Testing (外部设备测试)

在 μPAC-7186E 的串口 COM2 上连接 I-7000 系列模块,并通过 VxComm Utility 将 μPAC-7186E 上的 COM2 虚拟成本机 PC 的 COM10。这样,即可以通过应用程序 DCON Utility 访问 COM10 检索 到 I-7000 系列模块。

注意:应用程序 DCON Utility 须安装于主机 PC 中。该软件安装程序可通 过以下方式获得:

- http://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\_utilit y/setup/
- CD:\Napdos\Driver\DCON\_Utiltiy\Setup\

步骤 1: 执行 DCON Utility.

![](_page_35_Picture_7.jpeg)
步骤 2: 按如下图方式建立连接:



步骤 3: 单击按钮 COM Port 选择串口号,波特率及校验位。如: COM10, 115200, 19200, 9600, No-Checksum. (以 I-7000 系列模块设置 为例)



	🦉 Select th	e COM Port a	nd Baud Rate	. 🛛 🔀	
步骤 1	COM to s	earch:	- Time Out Se	tting :	
	COM10	-		500 <sup>ms</sup>	
	Baud Rat	e to search:-			
	921600	460800	230400	▼ 115200	
	57600	☐ 38400 ☐ 2400	✓ 19200	9600	
	4000	Select All	Clear	1	
				-	
	- Select Pr	otocol Option		DTU	
	Colort Ch	IV DCON		us HTU	
	- Select Ch	Disable	🗖 Enabl	e	L
				步骤2	2)
			$\checkmark$		
		<u>C</u> ancel	<u> </u>		
	-				
步骤 4: 单击图标 🕒					
<b>Ø</b> D	CON_UTILI	TY_VER [443]	The Found Ou	t I-7000/800	0 modu
<u>F</u> ile	COM Port S	earch <u>R</u> un <u>T</u> e	erminal <u>H</u> elp		
<u></u>					Star
Mod	ule 🛛 🗛	ddress Baud	rate Checksum	Status	
			:		

步骤 5: 若 VxComm 驱动正常工作,则 DOCN Utility 可访问 µPAC-7186E 的串口 COM2 的连接。



### 3.5.2 常见问题解答

问:由 VxComm 驱动创建的虚拟串口不能正常通讯。

**解:** 检测电源、串口连接线及网线是否正常连接,μPAC-7186E 网络 IP 地址、子网掩码、网关是否正确。

问:仍然不能正常访问虚拟后的串口。

解:步骤 1: 右键单击"我的电脑",在下拉菜单中选择"管理"选项。



步骤 2: 在"管理"对话框中选择"设备管理器"。



步骤 3:从"查看"的下拉菜单中选择"显示隐藏的设备"。





#### 步骤 4: 选择非即插即用驱动程序/ Ynsernet 选项。.

### 步骤 5: 选择 Ynsernet 项目,并单击鼠标右键查看相关属性



步骤 6: 检查是否提示有信息 "这个设备运转正常"。若显示该设备 目前不在正常运转之下,建议将该驱动移除并重新安装。

Ynserne	t Properties		?×				
General	Driver Details						
$\diamond$	Ynsernet						
	Device type:	Non-Plug and Play Drivers					
	Manufacturer:	Unknown					
	Location:	Unknown					
Devic	ce status						
This	device is working pr	operly.					
lf yo start	If you are having problems with this device, click Troubleshoot to start the troubleshooter.						
		Troubleshoot					
Device	usage:						
Use th	is device (enable)		~				
		ОКС	ancel				

- 问:程序可正常开启虚拟串口,但不能访问连接设备。
- **解:** 检测该设备电源及连接方式 (RS-232: RXD, TXD; RS-485: D+, D-; GND)。

# 4. 以太网 I/O 应用

# 4.1 Xserver 架构原理

典型的 TCP/IP 机制的标准对于一般的软件工程师来说比较复杂,可能需要花费大量的时间来开发 TCP/IP 协议的应用。

该架构命令协议专为 TCP/IP 系统设计且无任何限制,因此每一位软件工程师可自定义特殊协议而无须用户预定标准。同时,可能会产生如下问题:

- 该协议可靠吗?
- 该协议适用于所有需求吗?
- 以该协议创建的程序易于维护吗?
- 什么时候推向市场?
- 设计及调试该协议的工程成本多少?

### Xserver 架构可有效的解决以下问题:

- > Xserver 以独特的创新技术为用户提供可靠而稳定的解决方案。
- ▶ 该协议适用于所有 µPAC-7186E 系列需求。
- ▶ 该协议完全开放易于扩展,可有效降低用户开发费用。
- ▶ 可为用户应用提供一套易于使用的平台。
- ▶ 该协议提供完整的设计和维护标准,易于使用。

### Xserver 构架具有如下特点:

- Xserver 及 VCommnn.EXE 作为嵌入式软件专为 µPAC-7186E 系列 量身定制。
- ▶ 支持虚拟串口应用。
- ▶ 支持以太网 I/O 应用。
- ▶ 支持 I/O 扩展总线。
- ▶ TCP/IP 协议及命令协议完全开放,易于扩展。
- ▶ 提供易于使用的通用平台,便于开发应用程序。

在 Xserver 构架下,软件工程师几乎可在一天内开发出一套完整的应用。此外,我们将提供大概 50 ~ 100 套典型真实案例以供用户参考。从 大量的例程中,软件工程可迅速上手设计程序,大大缩短开发周期。更 多详情请参考章节 4.3。

# 4.2 Xserver 命令协议

## 4.2.1 IP 及端口配置

在开发以太网 I/O 应用前,必须详细了解相关 IP 地址及以太网端口号。 µPAC-7186E 及 µPAC-7186E 上所有的串口均使用相同的 IP 地址,仅 网络端口号有所不同。相关网络端口清单如下:

功能	IP 地址	端口号
Modbus TCP	192.168.255.1	502
Virtual 7000 (I/O 板卡)	192.168.255.1	9999
µPAC-7186E 配置	192.168.255.1	10000
μPAC-7186E: COM1	192.168.255.1	10001
μPAC-7186Ε: COM2	192.168.255.1	10002
μPAC-7186E: COM3	192.168.255.1	10003
μPAC-7186Ε: COM4	192.168.255.1	10004
μPAC-7186E: COM5	192.168.255.1	10005
μPAC-7186E: COM6	192.168.255.1	10006
μPAC-7186E: COM7	192.168.255.1	10007
μPAC-7186Ε: COM8	192.168.255.1	10008

**192.168.255.1** 为 µPAC-7186E 默认出产 IP 地址,也可根据实际需求进行修改。而于此相反的,各功能的对应的端口号则是锁定不可修改。

# 4.2.2 Xserver 命令台

命令	说明	命 令 (发送给 uPAC-7186E)	返回值 (返回绘主机 <b>PC</b> )
00[Arg]	停止模块执行 Vxcomm.exe 或 Xserver,并返 回 MiniOS7 控制台模式 Arg: "quit" 回返: • 成功: OK	00quit	OK
01	读取当前 Vxcomm.exe 或 Xserver 版本信息 返回: vm.n.rr[mm/dd/yy] • m: 主版本号 • n: 副版本号 • rr: 尾号	01	v3.2.25[06/21/2007]

02[Arg]	<ul> <li>设置 μPAC-7186E 串口波特率 (该设置并不储存于 μPAC-7186E 的 EEPROM 中)</li> <li>Arg: CBBBB <ul> <li>C: 串口号 (1~8)</li> <li>BBBB: 波特率 (1200, 2400, 4800, 9600, 19200, 57600, 115200)</li> </ul> </li> <li>返回: <ul> <li>成功: OK</li> <li>失败: Error</li> </ul> </li> </ul>	0219600 02219200	OK OK
03[Arg]	<ul> <li>设置 µPAC-7186E 数据格式</li> <li>(该设置并不储存于 µPAC-7186E 的 EEPROM 中)</li> <li>Arg: CDPS <ul> <li>C: 串口号 (1~8)</li> <li>D: 数据位</li> <li>(7, 8 → 适用于 COM1 及 COM2)</li> <li>(5, 6, 7, 8 → 适用于 COM3~COM8)</li> </ul> </li> <li>P: 奇偶校验位</li> <li>0, n → 无校验</li> <li>1, e → 偶校验</li> <li>2, o → 奇校验</li> <li>3, m → 标记校验 (恒为 1)</li> <li>4, s → 空格校验 (恒为 0)</li> <li>(0~2 → 适用于 COM1 及 COM2)</li> <li>S: 停止位</li> <li>(1 → 适用于 COM1 及 COM2)</li> <li>(1, 2 → 适用于 COM3~COM8)</li> </ul> <li>返回: <ul> <li>成功: OK</li> <li>失败: Error</li> </ul> </li>	0317N1 0328E1	OK OK
04[Arg]	<ul> <li>获得系统复位状态</li> <li>Arg: CC <ul> <li>CC 状态位(00~31)</li> <li>为客户端最多提供 32</li> <li>客户端调用该功能时必须使用不同的状态 位。当客户端查询状态痊时,系统将被复 位。这意味着,获得的状态位将是 0,直 到系统再次复位将显示 1。.</li> </ul> </li> <li>返回: <ul> <li>成功: OK</li> <li>失败: Error</li> </ul> </li> </ul>	0403	031

-			
05[Arg]	<ul> <li>设定 μPAC-7186E 串口 RTS 信号</li> <li>Arg: CS <ul> <li>C: 串口号 (1~8)</li> <li>S: RTS 状态. 0:禁用; 1:激活.</li> </ul> </li> <li>返回: <ul> <li>成功: OK</li> <li>失败: Error</li> </ul> </li> </ul>	0511	ОК
06[Arg]	<ul> <li>设定 μPAC-7186E 串口波特率</li> <li>(修改后设置将储存于 μPAC-7186E 的 EEPROM 中)</li> <li>Arg: CBBBB <ul> <li>C: 串口号 (1~8)</li> <li>BBBB: 波特率</li> <li>(1200, 2400, 4800, 9600, 19200, 57600, 115200)</li> </ul> </li> <li>返回: <ul> <li>成功: OK</li> <li>失败: Error</li> </ul> </li> </ul>	0619600 06219200	OK OK
07[Arg]	<ul> <li>设定 µPAC-7186E 数据格式</li> <li>(修改后设置将储存于 µPAC-7186E 的EEPROM 中)</li> <li>Arg: CDPS <ul> <li>C: 串口号 (1~8)</li> <li>D: 数据位</li> <li>(7, 8 → 适用于 COM1 及 COM2)</li> <li>(5, 6, 7, 8 → 适用于 COM3~COM8)</li> </ul> </li> <li>P: 奇偶校验位</li> <li>0, n → 无校验</li> <li>1, e → 偶校验</li> <li>2, o → 奇校验</li> <li>3, m → 标记校验 (恒为 1)</li> <li>4, s → 空格校验 (恒为 0)</li> <li>(0~2 → 适用于 COM1 及 COM2)</li> <li>S: 停止位</li> <li>(1 → 适用于 COM1 及 COM2)</li> <li>(1, 2 → 适用于 COM3~COM8)</li> </ul> <li>返回: <ul> <li>成功: OK</li> <li>失败: Error</li> </ul> </li>	0717N1 0728E1	OK OK
08[Arg]	<ul> <li>设定模块 IP 地址。当 IP 设定成功后,客户端 程序必须重新建立与 μPAC-7186E 的通讯连接</li> <li>Arg: iiipppIIIPPP <ul> <li>lii/ppp/IIII/PPPP: 3 位数</li> <li>(000~555)</li> </ul> </li> <li>返回: <ul> <li>成功: OK</li> <li>失败: Error</li> </ul> </li> </ul>	08192168255001 08192168254012	ОК ОК

	接收超时		
09[Arg]	<ul> <li>Arg: pttttt</li> <li>p: 串口号 (1~8)</li> <li>tttt: 0~99999, 单位 ms. 若设置为 'a', 'A' 或'-1',则 ttttt 将由系统侦到</li> <li>若触发电平&gt;1 (参考相关命令 32),则 ttttt 为 (trigger -1+3)=时间 (触发电平+字节) +1ms.</li> <li>若触发电平=1,则 ttttt 为 1 字节时间+1ms (最小值为 2ms).</li> <li>返回:</li> <li>成功: pttttt</li> </ul>	093	ОК
	• 失败: Error		
10	以以快兴名称 Arg: (none)	10	7186EX
	返回:模块名称		
11[Arg]	Arg: 任意字符串 (字符串长度最多 80 字节) 返回: • 成功: 字符串 • 失败: Frror	11TEST	<11TEST>
12[Arg]	设定模块网关地址 Arg: iiipppIIIPPP • lii/ppp/III/PPP: 3 位数(000~255) 返回: • 成功: OKReconnect. • 失败: Error	12192168000001	OKReconnect
13	读取模块网关地址 Arg: (none) 返回: • 成功: 网关地址 xxx.xxx.xxx.xxx (xxx: 0~255)	13	192.168.0.1
14[Arg]	设定模块 IP 掩码地址 Arg: mmmMMMmmmMMM • Mmm/MMM: 3 位数 (000~255) 返回: • 成功: OKReconnect • 失败: Error	14255255000000	OKReconnect

	读取模块 IP 掩码地址		
	Arg: (none)		
15	<u>쓰</u> 며.	15	255.255.0.0
	● 成功. IF 1吨均地址 vvv vvv vvv vvv (xxx: 0~255)		
	Arg: C • C: 电口号(1~8)		
16[Arg]		162	9600,N,8,1
	返回:		
	• 成功:波特率,数据位,校验位,停止位		
	EX:9600,N,8,1		
	读取 µPAC-7186E 数字量输入信号		
	Arg: pppp		
	● pppp: 4 位 16 进制数, DI 的输入地址.		
17[Arg]	如:0300 即为输入地址为 0x300	170003	55
	返回:		
ĺ	● 成功:显示 DI 数据		
	<ul> <li>● 大火: Effor</li> <li>□DAC 7196E 生送粉 字 是 給 山 粉 握</li> </ul>	<u> </u>	
	μFAC-/ 100E 及达效丁里間山效加		
	Arg: ppppdd		
	● pppp:4 位 16 进制数,DO 输出 DO 地址		
18[Arg]	如:03f8 即为输出地址 0x3f8	1803f855	ок
- L U.	• dd:2 位 16 进制数,输出值		
	返回:		
	● 成功: OK		
	•_ 失败: Error		
	用户函数分路指令		
19[Arg]	Arg: 用户定义变量	19(用户定义)	(用户定义)
	● 返凹用尸汉疋徂 谢活/埜田 5 位 7 段 Led 显示信息		
	Arg: C		
	● c=1:激活 5 位 Led		
20[Arg]	● c=0:禁用 5 位 Led	201	ОК
	返回:		
	● 成功: OK		
	• 失败: Error		
	读取模块 MAC 地址		
	Ara: (none)		
21	, "9. ()	21	71:88:03:3b:02:c7
	返回:		
	• xx:xx:xx:xx:xx:xx,xx: 16 进制数值		

22	读取模块 MiniOS7 版本号 Arg: (none) 返回: vm.n.rr [mm/dd/yy] • m: 主版本号 • n: 副版本号 • rr: 尾号	22	
23[Arg]	用户函数分路指令,包含有 socket 号和数据长度。 Arg:用户定义变量 返回: •返回用户设定值	23(用户定义)	(用户定义)
24[Arg]	设定返回指令号 Arg: m ● m=0: 反馈不添加指令号 ● m=1: 反馈添加指令号 返回:	240 241	ОК 240К
25[Arg]	<ul> <li>查询系统参数</li> <li>Arg: m</li> <li>m='S': 查询系统超时值 (/ST)</li> <li>m='W': 查询 socket 超时值 (/W)</li> <li>m='N': 查询 socket 最大序号</li> <li>m='N': 查询工作模式选项:/M0 或 /M1</li> <li>m='B': 激活/禁用广播模式</li> <li>B1:激活广播模式,</li> <li>B0: 禁用广播模式(默认:B0)</li> <li>m='E': 查询返回模式指令</li> <li>E1:激活,E0: 禁用 (默认:E0)</li> <li>m='L': 查询 7 段 LED 状态,</li> <li>L1: 激活 LED 显示, L0: 禁用 LED 显示.</li> <li>(默认: L1)</li> <li>m='I': 查询 µPAC-7186E 网络</li> <li>NAME/IP/MASK/GATEWAY/MAC</li> <li>返回:</li> <li>成功: mxxx. m: 'S', 'W', 'N', 'M' xxx: 数值</li> <li>失败: Error</li> </ul>	25M 25S	M0 S0

	识白互依杂粉		
	<b>反</b> 定 杀 统 麥 级		
	Arg: mn		
26[Arg]	<ul> <li>m='SAVE': 设定系统参数是否将保存 EEPROM 中,若不设定该参数,默认值为 SAVE=0.</li> <li>例: "26SAVE=1", "26SAVE=0" m='M': 设定工作模式 /M (n:0 n:1)</li> <li>m='B': 设定广播模式("26B0") ("26B1")</li> <li>m="S": 设定系统超时 /ST ("26S300000")</li> <li>m="W": 设定 socket 超时/W ("26W8640000")</li> <li>m="E": 设定返回模式指令</li> <li>m="I": 设定返回模式指令</li> <li>m="I": 设定 IP, MASK, GATEWAY, 或 MAC (IP/GATEWAY/MASK/MAC 必须大写)</li> <li>设定参数,使用逗号隔离 如: "26SAVE=1,M=0,B=0,S=300000"</li> <li>返回:</li> <li>成功: m=n. m='SAVE', 'B', 'S' or 'M', n=设定值</li> <li>失败: Error</li> </ul>	[in]: 26SAVE=1,M=1,B=0 [in]: 26UDP=1	[out]: SAVE=1,M=1,B=0 [out]: UDP=1
	送出暂停信号( Break ). Arg: CM • C: COM port (1~8)	2711	OK
27[Arg]	• M: 1 激活; 0: 禁用	2710	ОК
	返回: • 成功: OK • 失败: Error		
28[Arg]	<ul> <li>设定/查询流控制的 CTS/RTS 模式</li> <li>Arg: p[mn] <ul> <li>p: 串口(only support COM1/3/4/5)</li> <li>m, n: m 设定 CTS 模式, n 设定 RTS 模式</li> </ul> </li> <li>'0': 禁用 <ul> <li>'1': 激活. (RTS 必由用户程序控制)</li> <li>'2': 激活, 由硬件自动控制</li> <li>'3': 激活, 由硬件自动控制(库).</li> <li>对于 CTS, 模式 '1' 和 '3'相同若指令不能添加参数 mn, 转值为初次确定值</li> </ul> </li> <li>返回: <ul> <li>成功: 返回刚刚输入的数值</li> <li>失败: Error</li> </ul> </li> </ul>	28111 281	111 111

29[Arg]	<ul> <li>查询 CST 状态</li> <li>Arg: p <ul> <li>p: 串口 (仅支持 COM1/3/4/5)</li> </ul> </li> <li>返回: <ul> <li>成功: CTSn=0 or CTSn=1. n: 串口</li> <li>失败: 返回 "No CTS n SIGNAL" n: 串口</li> </ul> </li> </ul>	291 292	CTS1=0 CTS2=1
30[Arg]	发送字符串及接收与刚刚发送字符串相同的数据 字符串包含 30, 无论指令号是否激活 (240 或 241). Arg:ssss • ssss: 任何字符串. (最多可长达 1460 字节) 返回: • 成功: 返回输入字符串 • 失败:	[in1]: 30123456789 [in2]: 30	[out1]: 30123456789 [out2]: 30
31[Arg]	<ul> <li>设定芯片 16550 下的串口触发电平。 (仅仅支持 COM3~8, COM1~2 不适用).</li> <li>可选参数: 1,4,8,14 (系统默认值为 8)</li> <li>Arg: p [LL]</li> <li>p: 串口号 3~8。</li> <li>[LL],"1","4","8","14"4 个参数选择其一.</li> <li>若指令不能添加参数 LL, 将返回初次确定值</li> <li>返回:</li> <li>成功: return the value that has just been input back, Format30p[L[L]]</li> <li>失败:</li> </ul>	[in1]: 31314 [in2]: 313	[out1]: 314 (Com3,triggerlevel=14) [out2]: 38 (Com3,triggerlevel=8)
32[Arg]	<ul> <li>设定串口缓冲触发电平 (字节),即为接收与触发 电平相匹配的数值,返回缓冲中数据</li> <li>该指令接收传输数据长度一定的数据更为有 效,而对于需要每次等待返回超时信号的数据 就不是很适用。</li> <li>Arg: p [nnn] <ul> <li>p: 串口 1~8</li> <li>nnnn: "0"~"1460"</li> <li>返回之前设定的数值,而在变数中不添加 任何 nnnn</li> </ul> </li> <li>返回: <ul> <li>成功: 以格式 31pnnn 返回刚刚输入的数 值</li> <li>失败: Error</li> </ul> </li> </ul>	[in1]: 321 [in2]: 3221460	[out1]: 10 [out2]: 21460

33[Arg]	<ul> <li>Arg:</li> <li>"33dir\n", "33dir a\r" → dir a</li> <li>"33dir b\r" → dir b</li> <li>"33load a size\r" → 询问下载文件 (1)</li> <li>"33+FILE DATA" → for download file (2) Items (1) and (2) run to update the files need the assistance by program.</li> <li>"33del a\r" → 删除 a 分割区的文件</li> <li>"33del a\r" → 删除 b 分割区的文件</li> <li>"33reset\r" → 重启 7188e(by WDT)</li> <li>"33fileno a\r" → 读取 a 分割区文件数与剩 余空间</li> <li>"33fileno b\r" → 读取 b 分割区文件数与剩 余空间</li> <li>[返回:</li> <li>成功: 命令回复</li> <li>Fault:</li> </ul>	[in]: 33fileno a [in]: 33del a	[out]: 1 99286 [out]: Delete file(s) in disk A OK.
34	<ul> <li>查询 Vcomnn.exe 的库文件版本号与日期,</li> <li>包含 7186el.lib 和 tcpipl.lib</li> <li>Arg: (none)</li> <li>返回:</li> <li>成功: 7186el.lib Ver. n.nn[cc cc cccc], tcpipL.lib Ver. n.nn[cc cc cccc] n.nn: 版本号, [cc cc cccc] : 日期</li> </ul>	34	7186EL.LIB Ver. 2.3[Jun 21 2007], TCP_DM32.LIB Ver. 1.18[May 04 2007]
35[Arg]	清空串口输出/输入缓冲区 Arg: ptr ● p: port ('1'-'8') ● t: '1' 清空 tx 输出缓冲区 ● r: '1' 清空 rx 输入缓冲区 返回: none	35111	
36	查询 timeticks Arg: none 返回: timeticks value	36	TIMETICKS=4219529
37[Arg]	查询 MSR (Modem status register) 状态 Arg= p • p: COM port, 1~8:COM1~COM8 返回: • MSR 状态值	371	MSR1=30

#### 注意:

- 在括号 ()中的数值即为相关参数所占空间大小 (字节)。
- 在参数中间请勿加入空格 (用户自定义命令除外)。
- 所有命令 (用户自定义命令除外) 响应均以 CR(0x0d)作为终止符。
- 更多有关 Xserver 命令协议及相关参数请参考 vxcomm.htm。 (http://ftp.icpdas.com/pub/cd/8000cd/napdos/7188e/tcp/vxcomm/d oc/)

## 4.2.3 命令行操作

命令行操作						
Vcomnnnn.exe [/Option] (nnnn 为库文件 Vxcomm.lib 版本号)						
操作	说明	备注				
/ <b>X</b>	μPAC-7186EX					
	Multi-echo 模式					
/MO	从 µ PAC-7186E 各串口得到的返回数据将直接送达每个连					
	接到 µ PAC-7186E 的客户端上					
	(μ PAC-7186E 默认模式)					
	Signle-echo model.					
/M1	从 µ PAC-7186E 各串口得到的返回数据将直接送达连接到					
/	μ PAC-7186E 的指定客户端上					
	(Xserver 默认模式)					
	Socket 超时					
	socket 超时值决定系统在放弃操作前将耗费多长时间等待					
	接收下一个封包。若超时期满,则 Vxcomm.exe/Xserver					
/Wxxx	将自动释放网络连接。					
	xxx: 超时 (时间单位: 杪)					
	默认:0					
	xxx=0: 禁用/W 操作					
	位定 µ PAC-7186EX 接收封包间隔的系统超时值。					
	右超时时长期满,则 Vxcomm.exe/Xserver 将日动重启系					
/STxxx						
	XXX: 超的 (的 间 单 位: 秒)					
	反正 PAC-/100EX 问申口及达证令间隔超的值。石超的时 上期进一则 Viceomm ave Veenver 收益会该数据					
	大期俩,则VXCOMM.eXe/AServer 将风开该数据。	适用于模式 M1				
/ 1 X X X	XXX: 超时 (时间中位: 位)	(single-echo 模式)				
	从人:0					
	XXX-U. 示用/I 珠中 					
/FCnnnnn	以定 UIS 狂耐侠八。 nnnn 即为 COM1-5 久个串的对应店					
	n· 0~3 再名详情语发老命公参数 "28"					
	日、0 5 Cシド間間参写曲マ参数 20 投完 RTS 控制構式					
/FRnnnn	的 <b>nnnn</b> 即为 <b>COM1-5</b> 各个串的对应值。					
	n· 0~3 更名详情请参考命令参数 "28"					
	出。 设定结止符					
	当收到数据的最后一个字符时将核对结止符。接收到的数据					
/Fhh	将立刻返回。					
	Hh 为 16 进制数,默认值为 0					
	Ex: 设 0x0A 为结止符"/E0A":					
	设 0xFF 为结止符"/EFF"。					
	0 表明没有结止符					
		1				

	改变命令端口号(默认为 10000)。	
/Pxxxxx	若命令端口号改为 200,则 TCP/IP 映射对应串口的端口号	
	将为 201~208.	
191	命令端口及各个串口映射的 TCP/IP 端口将仅仅只允许建立	
/31	一个 TCP/IP 网络连接,任何其它连接均为非法。	
// 0	执行 "L0"命令将可控制 5 位 7 段 LED 是否显示。执行该命	
/LU	令将可对没有显示功能的模块节约运行周期。	
/Zxxx	该命令可用于设置 ACK 延时时长,默认为 1000ms。	
/DHCP=x	设定 DHCP 功能 /DHCP=0 (禁用), /DHCP=1 (激活)	
	设定 PING 功能。 /PING=nnn, 数值"nnn"为 PING 的频率,	
/PinG-nnn	推荐频率值大于 2。	
/P23=n	设定 Telnet 功能。 /P23=0 (禁用), /P23=1 (激活)	
/HTTP=n	设定 web 服务功能。 /HTTP=0 (禁用), /HTTP=1 (激活)	

#### /M0: Multi-echo 模式

应用 1: 一个客户端通过网络向 Xserver 发送请求, Xserver 将从设备 返回的数据发送到每个连接到 µPAC-7186EX 的客户端上。



应用 2: 无需客户端发送任何需求 Xserver 主动将任何返回数据传送到 每个连接到 µPAC-7186EX 的客户端上。



#### /M1: Single-echo 模式

应用 1: 某个客户端通过 Xserver 接连到设备, Xserver 接到请求后将 返回数据发送到指定客户端,而其它客户端接收不到任何数 据。



情况 2: 若无任何客户端发送请求命令给 Xserver, Xserver 即使获得数据也不会发送给任何客户端。





# 4.2.4 Xserver 构架流程图



虽然开发一套嵌入式以太网通讯程序将是一件很繁重的工作,但在使用 基于 Xserver 架构后,用户可方便快捷地完成一整套编程工作。因为用 户开发时仅仅只需修改我们早已设计好的在 User.c, vModbus 和 v7000.c 中的7个框架即可实现。这7框架有如下特点:

### UserInit (void):

Xserver 一旦执行该函数即表明 µPAC-7186EX 将正式启动。因此所 有初始值或初始状态必须在此函数内设定并声明。

### UserLoopFun (void):

Xserver 将在每个扫描周期内执行该函数。Xserver 扫描周期可以短时间完成应用,因此实时工作可放在该函数中执行。(参考 Demo12)

### UserCount (void):

该功能以函数 AddUserTimerFunction 设定的时间时隔触发。因此推荐用户在 UserInit 呼叫 AddUserTimerFunction 函数使 Xserver 初始 化 UserCount 功能。另:最长时间时隔为 65.535 秒(2^16–1 ms) (参考 Demo9)。

#### UserCmd (Cmd, Response):

当客户端程序程序向 Xserver 的 10000 号端口发送命令时, Xserver 将执行该功能,而在 UserCmd 函数命令的定义均可由用户自行设计。当 Xserver 从 10000 号端口收到数据时,将会自行检测数据。若数据以"19"作为首字符, Xserver 则检测通过并将"19"过滤接收用户其它数据(不包含 19)作为新的 UserCmd 的首参数。

用 户 通 过 UserCmd 自 定 义 命 令 协 议 。 如 : <19,Read/Write(1),address(4),[value(2)]>取代命令 17 和 18,一旦用 户发送命令 "19R03f8",系统将会读取地址为 03f8 的数据;发送命 令 "19W03f85a"则将写入 5a 到地址 03f8 之上。

用户也可判定是否在命令中配以其它参数。由于整个命令协议均由用 户自定义,故所有命令格式都可适用。

最后 UserCmd 将所有数据复制到响应参数 Response 中,由 XServer 通过 10000 号端口发送数据到客户端。(参考 Demo4)

### VcomUserBinaryCmd (TCPREADDATA \*p):

若客户端程序发送命令 "23"到 Xserver 的 1000 号端口, Xserver 将 执行该函数。因此, VcomUserBinaryCmd 函数功能与 UserCmd 类 似,一旦客户端程序发送命令 "23",该函数将自行接收 TCPREADDATE 类型信息。TCPREADDATE 声明格式如下:

Type define t\_TcpReadData{ Int Comport; int Socket; int Length; char\* ReadUartChar; } TCPREADDATA;

p->ReadUartChar: 命令数据保存缓冲 (包含 "23")

p->Length: 命令数据长度 (包含 "23")

p-Socket: 当客户端发送命令 "23"到 µ PAC-7186EX 时, Xserver 将 自动分配 socket 序号给客户端。因此,数据响时可返回到指定客户 端,而调用的 VcomSendSocket(int skt, char\* data, int cnt)首参数 须为 socket 序号。(参考 Demo23)

## VcomCmdModbus (TCPREADDATA \*p):

当 客 户 端 程 序 发 送 命 令 到 Xserver 的 502 端 口 时, VcomCmdModbus 将被启动。该函数采用 Modbus TCP 通讯协议与 各设备相连。(参考 Demo27)

### VcomCmd7000 (TCPREADDATA \*p):

当客户端程序发送命令到 Xserver 的 999 端口时,函数 VcomCmd7000 即刻执行。该函数用于与 7000 串行系列相兼容的扩展设备通讯。

# 4.3 Xserver 构架例程清单

用户在开发完基于 Xserver 架构的程序后,必须将其可执行程序下载到 µPAC-7186EX 中执行,以检测所有程序是否正常运行。

例程	函数	说明	客户端
XDemo04 (Original X-Server)	返回命令字符串	将用户自定义命令返回	Client1
XDemo05	返回数据到指定客户端	Xserver 将返回 "7188_Series." 到客户端	Client1
XDemo06	读/写 7188E 的 I/O 端口	该例程说明如何使用命令 19 取代命令 17, 18	Client1
XDemo07	使用 printCom1 调试程序	用 户 可 用 函 数 "printCom1" 透 过 μPAC- 7186EX 的串口 COM1 发送调试信息显示在 PC 显示器上	Client1
XDemo08	使用 Print 调试程序	用户可使用函数 <i>Print()</i> 透过 μPAC-7186EX 的 串口 COM1 将调试信息显示在 PC 显示器上。 若确定用户函数 <i>Print()</i> ,必先使用函数 <i>DisableCom()</i> 及 <i>RestoreCom()</i> 禁用函数 printCom1	Client1
XDemo09	计时器例程	UserCount() 将每秒自执行一次,且计数器将 在函数 UserCount()中累加,在计数器清零 后,PC 可通过读取计数值了解程序运行时间	Client1
XDemo10	刷新看门狗例程	若用户函数执行时长超过 1.6 秒,用户必须在 该函数内插入 <i>RefreshWDT()</i> 以刷新看门狗避 免操作系统重启	Client1
XDemo11	实时 I/O 操作	该例程说明如何在函数 UserLoopFun()中对 DI/DO进行实时操作	Client1
XDemo12	扫描周期应用	UserLoopFun()中计数器将在每一个扫描周期 是累加,在计数器清零后,PC 可读取该函数 执行次数。因此用户可使用该例程测试 Xserver 是否运行正常	Client1
XDemo13	脉宽测量	UserLoopFun() 读取 D/I 信号, 若信号改变, 则记录时间标签计算其信号宽度	Client1
XDemo14	控制 5 位 7 段 LED	函数	Client4
XDemo15	读取模块 7017/7018 通道信号	该例程阐明如何发送命令用于读取以连接于 μPAC-7186EX 串口 COM2 上的模块 7017 或 7018 上各通道数值	Client4
XDemo16	通过串口与其它设备通讯	该例程阐明如何用户串口与 7000/87K 系列模 块或其它串型设备通讯,而非使用 7000 专用 函数 ReceiveResponseFrom7000() 及 SendCmdTo7000()。且命令及各模块响应数 据也并非全由 0x0d 作为终止符。更多详情请 参考用户手册修改终止符一节	Client4
XDemo17	脉冲宽度测量	该例程与 XDemo13 类似,用户测量信号改变 及对应的脉冲信号宽度	Client4
XDemo18	读取 I-7000 系列模块 ID	该例程阐明如何与连到 μPAC-7186EX 上串口 COM2 的 7000 系列模块进行通讯	Client4

XDemo19	读取 64 位全球唯一硬件序列 号	唯一的硬件序列号常常可用于对用户程序进行 保护加密。用户可使用工作 7188xw.exe 进入 μPAC-7186EX 的操作系统 MiniOS7,即可获 得该序列号。用户可在 Xserver 程序之初加入 检测代码,以检测硬件序列号来决定是否继续 执行该程序.	Client4	
XDemo20	读/写/清除 NVRAM 可用数据类型 int, float, string.	由 NVRAM 特点所至,该存储介质响应时间 短,可无限次擦写且配置有后备电池可确保运 行长达 10 之久 函数 NVRAM_MultiRead()用于读取多路数 据,函数 NVRAM_MultiWrite()则用数据写入	Client4	
XDemo21	在 <i>UserCount()</i> 函数中控制硬 件	在函数 UserCount() 中进行硬件的操作是非法的。若用户需要在函数 UserCount()中操作硬件,推荐可使用状态标识符的方式处理,而在函数 UserLoopFun()中完成相应的操作。 另外在函数 UserInit(), UserLoopFun(), UserCmd().中硬件操作是合法的。该例程每秒执行函数 UserCount()一次进行数据的累加,并将对应的数据显示在5位7段 LED.	Client4	
XDemo22	数据采集,压缩及取平均值	为中小客户 ODM 服务	CheckValue	
XDemo23	返回所有数据(包含 "23")到指 定客户端	使用函数 VcomSendSocket 返回所有数据 (包含 "23") 到指定客户端	Client4	
XDemo24	使用倒计时定时器	系统拥有 8 个倒计时时钟 (通道 0 到通道 7). 该 例程使用通道 0. 倒计时定时器初始值为 1000 ms,一旦倒计时定时器值为 0, LED 对应的数 值便会累加	None	
XDemo25	数据采集, 压缩及取平均值	无法使用	CheckValue	
XDemo27	Modbus/TCP 到 Modbus/RTU Master/Slave 转换器	库文件 Modbus Lib 已经发布,有关 Modbus 更多信息请访问 CD:\Napdos\Modbus\*.*	任何 Modbus/TCP 客户端程序	
		在许多应用中,用户需要将一些关键信息存储 在纯文本文件中,任何第三方程序均可方便的 读取其中数据。用户可利用函数 FSeek 获得文 本中特定信息。		
		XDemo28 配置文件样品		
XDemo28	从纯文本文件读取信息(基础)	星型注释 Item1 Item2 Item3 Item4	Client4	
		* 1st 19200 on 1.25 ;comment1 * 2nd 9600 off -1.3 ;comment2 * 3rd 2400 off 19.34 ;comment3 = 设置结止 =		

XDemo29	从纯文本文件读取信息(高级)	在许多应用中,用户需要将一些关键信息存储 在纯文本文件中,任何第三方程序均可方便的 读取其中数据。用户可利用函数 <i>GetProFileInt()及 GetProFileStr()</i> 获得文本中 特定信息。 	Client4
		·····································	
XDemo30	I/O 扩展总线:X304	X304: 3 路 12 位 A/D 通道 (+/- 5.0V) 1 路 12 位 D/A 通道 (+/- 5.0V) 4 路 D/I 通道 4 路 D/O 通道	Client4
XDemo31	I/O 扩展总线: X310	X310: 2 路 12 位 A/D 通道 ch0: 0~20mA ch1: 0~10V 2 路 12 位 D/A 通道 (0.0~10.0V) 3 路 D/I 通道 3 路 D/O 通道	Client4
XDemo32	后备系统 (适用于 Demo Board I)	当客户端连接到 μPAC-7186EX 时,该客户端 即可获得系统的控制权。 一旦客户端与 μPAC-7186EX 失去连接,或客 户端程序崩溃,μPAC-7186EX 可继续以用户 前先定义好在 Xserver 中的控制方式进行操 作,以保证系统可以继续正常运行。 以 Demo Board I (May/2002)为例: μPAC-7186EX (COM2, Baudrate=9600, checksum disable) + I-7012(#01, Al*1, 0~20mA) In <==> IOut of I-7021 +I-7021(#02, AO*1, 0~20mA) IOut <==> Current meter <==> In of I-7012 +I-7021(#02, AO*1, 0~20mA) IOut <==> Button0~Button3 Button0 ==> 灯 Button1 ==> 步进电机 Button1 ==> 步进电机逆转 DO0~DO3 <==> A+/A-/B+/B- 步进电机 向位 DO4~DO7 <==> 灯 0~灯 3 + I-7013(#04, RTD*1) <==> RTD	
XDemo34	使用 "命令 23" 每秒读取计数 器数值	该例程与 XDemo23 类似,用户可使用全量变量保存 socket 序号,然后可使用在其它子程序中的 VcomSendSock() 函数发送数据到指定 socket 连接	

XDemo57	µPAC-7186EX 客户端函数	该例程为 μPAC-7186EX 提供客户端函数,用 户也可使用该函数与其它 μPAC-7186EX 通讯 另外,μPAC-7186EX 也可作为 TCP/IP 通讯的 客户端或服务器,而无需任何 PC 支持。	Client1
XDemo62	读/写 EEPROM	EEPROM 共计大小 16K 字节。该存储介质适 用保存诸如:用户 ID、密码等不频繁更改的数 据。且 EEPROM 可重复擦写高达 100 万次。	Client4
XDemo63	读/写/清空 Flash	<ul> <li>μ PAC-7186EX 系列拥有两种类型 Flash 记忆体规格,其区别主要在于 Flash 记忆体大小有别,一种是 256K,另一种则为 512K。两者均可重复擦写达 10 万次。</li> <li>利用函数 <i>Flash_MultiRead()</i> 读取数据;通过函数 <i>Flash_MultiWrite()</i> 将数据写入 Flash.</li> </ul>	Client4
XDemo64	SMTP 客户端(发送 E-mail)	该例程将阐明如何在 μPAC-7186EX 中使用简 单邮件传输协议 (Simple Mail Transfer Protocol: SMTP)透过 SMTP 服务器发送邮 件到用户指定电子邮箱中	Client4
XDemo65	使用 ICMP 协议进行时间同步	该例程用于μPAC-7186EX 为用客户端或服务 器通过 ICMP 协议与其它服务器通讯实现与基 准时间同步。	Client4
XDemo69	保存 7018 及 DI 通道数值,并 储存到 Flash 中 (Log)	该例程阐明如何在 µ PAC-7186EX 系列中建立 数据记录日志系统。已保存在 Flash 中的有效 日志总量为 12288,另外每个日志占用 Flash 空间大小为 16K 字节	Client4
XDemo70	访问域名地址	在许多应用中,用户希望通过域名而非 IP 地址 连接到服务器。该例程将阐明如何在 μ PAC- 7186EX 中使用域名系统。 获得域名的 IP 地址可使用函数 DN_to_IP().	None

注意:更多例程信息请参考文件 xserver.htm 及 Function.htm http://ftp.icpdas.com/pub/cd/8000cd/napdos/7188e/tcp/xserver/.

# 4.4 基于 Xserver 构架程序设计

开发基于 Xserver 架构的程序, 需要下表九个文件:

项目	文件	路径
头文件	7186e.h, TCPIP32.h, Vxcomm.h	
库文件	7186el.lib, TCP_dm32.Lib, Vcomnnn.lib	7186E\TCP\Xserver\Demo\BC\Lib
用户源文件	User.c, vModbus.c, V7000.c	7188E\TCP\Xserver\Demo\BC\Demo4

注意: 库文件 VcomNNNN.lib 中 NNNN 为该库文件版本号。

## 4.4.1 Xserver 初始源码

所有 **user.c** 文件均可分为两部分: 一、头部的程序说明;二、程序主体。

● 头部说明:

/* DEMO4: Echoes comma Compiler: BC++ 3.1 ,TC- Compile mode: large Project: user.c	and string ++ 3.0, TC++ 1.01, TC 2.0	工程文件内容
v7000.c vModbus.c \Lib\7188EL.Lib \Lib\TCPIPL.Lib V.ib\VcomNNNN	[after Vcom_3002]	version
	LLD, with NNNN being the lib life's v	version.—
19~!@#\$ -> Any non-n	ull command will be accepted.	客户端程序使用该通 讯协议与 Xserver 通讯
This demo is the origin User can modify their o	nal user.c own Xserver from this file.	
Hardware: 7188E	硬件设备清单	该例程说明
Refer 7188e\TCP\Doc\[B 7188e\TCP\Xserver\ 7188e\TCP\Xserver\ to get more information [17/Nov/2001] by TCK — */	Big5/Eng/Gb2312]\Vxcomm.htm Xserver.htm Function.htm ·     尾部修改信息	

```
▶ 程序主体:
#include <string.h>
#include "...Vib\7188e.h"
#include "..Vib\vxcomm.h"
void UserCount(void)
{
    // user's timer trigger function
    // In this function, user cannot use any function that will use the hardware signal
"clock",
    // Such as:
    // 1. ClockHigh(),ClockLow(), ClockHighLow(),
    // 2. Any EEPROM functions.
    // 3. Any 5DigitLed functions.
    // 4. Any NVRAM function.
    // 5. Any RTC function.(GetTime(),SetTime(),GetDate(),SetDate())
    // Refer to demo9 for example code
}
void UserInit(void)
ł
    // User's initial function
    // Timer initialized for UserCount()
    // I/O or variables initialized for UserLoopFun()
    // I/O or variables initialized for User's functions in this file
    // Refer to demo9 & demo11 for example code
}
void UserLoopFun(void)
{
    // VxComm.exe will call this function every scan time
    // Refer to demo11 for scan-time evaluation
}
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
    // User's command interpreter
    // Refer to all demos
                           /* Not Null command */
    if (Cmd[0])
    Ł
        strcpy(Response,Cmd); /* echo user's command back */
                            /* return OK */
        return 1;
                  /* return ERROR */
    return 0;
}
```

int VcomUserBinaryCmd(TCPREADDATA *p)
/* VXCOMM.EXE 2.6.12(09/04/2001) or later will support this function.
TCP PORT 10000, command 23 will call this function.
user can get the following message:
p->ReadUartChar : the buffer store the command data(include "23")
p->Length : the command data length(include the two byte "23")
<i>p-&gt;Socket : the socket number that receives the command, that is, when the user</i>
function wants
to return a message to the client, just use the socket to send data. use: VcomSendSocket(p->Socket.pdata.datalength):
*/
<i>return 1; /* any value will be accept */</i>
}

注意: 详情请参考

http://ftp.icpdas.com/pub/cd/8000cd/napdos/7188e/tcp/vxcomm/doc/ to get new version information.

### 4.4.2 推荐使用 BC++ 3.1 编译程序

C 语言编译器以其独特设计理念,与多种底层硬件相兼容,因此常常用 来开发各种应用,而常用其中常用的编译器有如下几款:

- BC++ 3.1~5.02
- TC++1.01
- TC 2.01
- MSC
- MSVC++ (Prior to version 1.52)

**注意:**免费版的 Turbo C 2.01 及 Turbo C++ 1.01 编译器可从 Borland 公 司网站获得。

网址: http://dn.codegear.com/museum/

ICP DAS(泓格科技)建议使用 BC 3.1 作为创建工程文件及程序编译器的工具使用。在使用该程序进行编译前,请先核实如下项目:

- 输出标准的 DOS 可执行程序
- 设定 CPU 为 80188/80186
- 在需要进行浮点运算时,设定浮点 EMULATION (且该工程应用 中,不可选用 8087)。
- 取消会使程序增大的调试信息功能(MiniOS7 支持该功能)。

BC++编译器使用方式如下: 步骤 1: 执行 Borland C++ 3.1



步骤 2: 创建源文件(\*.c) 步骤 2.1: 从菜单 File 中选择 New

	ommand	Prompt	- bc							- 🗆 🗙
Ξ	File	Edit	Search	Run	Compile	Debug	Project	<b>O</b> ptions	Window	Help
	New									
	Save Save Save	as all	F2							
	Chan Prin DOS	ge dir t shell								
	Quit		Alt+X							
۲ <sup>ι</sup>	]				——— Me	ssage =				:1=[`]=ŋ ▲
										1
										- T
<b>F1</b>	Help	Creat	e a new	file	in a new	Edit wi	ndow			

步骤 2.2: 输入如下代码,注意代码格式

#include "7186e.h"	
void main(void) {	
<pre>InitLib(); Print("Hello world!\r\n");</pre>	
}	

步骤 2.3: 单击 **File** 菜单中的 **Save**,并输入文件名称 **Hello.C** 存储文件。



步骤 3: 创建新的工程文件 (\*.prj), 单击 Project 菜单中 Open project, 输入工程名称

C K	ommand Prompt - ex	it			- 🗆 ×
=	File Edit Se	arch Run C	Lie hug	<b>Project</b> Options	Window Help
#inc	lude "7186E.h"			Open project	
void	l main(void)			orose project	步骤 1
s (	[[[]]]	= Open Project	File ====	tem	
>	Upen Project	t File		files	
	Hello.rko	<b>生曜</b> 2		<b>生曜</b> 3	
	••••	J 34K Z		J SK J	
					تح
			Can	cel _	1
			Hel		
	C:\7188E\*.P	a.			
<b>P</b> 4	DEM091	Directory Ja	n 17,2006	5:54pm	

步骤 4: 添加工程文件必要文件 步骤 4.1: 选择源文件,并单击按钮 Add 进行添加



步骤 4.2: 选择库文件,并单击按钮 Add 添加,点击 Done 完成



步骤 5: 在菜单 Options 中,单击编译器进入菜单,并选择 Code generation



步骤 5.1:修改存储模型为大型



步骤 6: 在菜单 Options 中,单击编译器进入菜单,并选择 Advanced code generation.


步骤 6.1: 设定 Floating Point (浮点类型)为,指令集为 80186

[1] Advanced Code Generation —————
Floating Point (•) Fmulation (•) Fmulation (•) 80287/387 [X] Debug info in OBJs [X] Prowser info in OBJs [X] Fast floating point
Instruction Set + 8000,0000 步骤2 + 80186 + 80186
→ 80386 Far 步骤 3 101d 32767
OK Help

步骤 7: 在菜单 Options 中,单击 Entry/Exit Code 设置输入/输出代码生成器



步骤 8: 从菜单 Options 中,单击 Debugger 进入调试器设置



步骤 9: 从菜单 Options 中,单击 Directories 进入 BC++ 3.1 包含文件 及库文件目录选项。默认目录设置与 BC++ 3.1 程序目录一致。

ca Command Prompt - bc		- 🗆 X
😑 File Edit Search Run Compile Debug Project 이	ptions Window	Help_
<pre>#include "7186E.h" void main(void) {     InitLib();     Print("Hello world!\r\n"); }</pre>	Application Compiler Transfer Make Linker Librarian Directories	▶ 步骤 1
步骤 2 Include Directories D:\BC31\INCLUDE D:\BC31\LIB utput Directory 步骤 3 ource Directories 步骤 4 Kmmvel Help	nent Code D 22 n∕a utput file	▶ -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 -2

注意: Include Directories(包含文件目录)内为标准包含文件, Library Directories (库文件目录)内为 BC++ 3.1 启动目标文件及运行库文件。

步骤 10: 从菜单 Compile 菜单,单击 Build all 生成工程



**注意:**更多有关编译器设置详情请参考文件 compiler.pdf。 http://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/document/

## 4.5 基于 Xserver 构架例程

**Demo5**:返回指定字符串给客户端 主体程序部分代码:

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
    // user's command interpreter
    // refer to all demo
    strcpy(Response,"7188_Series."); // return "7188_Series." string to clients
    return 1; // return OK
}
```

## Demo6: 读/写 µPAC-7186E 的 I/O 通道

头部说明代码:

```
// 19i n, 19l n -> read D/l value (hex) from address n(hex)
// 19o add val, 19O add val -> output D/O value (hex) to address n(hex)
// 19 o add val, 19 O add val
// This demo shows how to use command 19 to replace command 17, 18
// Hardware: [7188EA] or [7188EX + D/l/O expansion board]
// Note: Base address of 7188EA's DI is 1.
```

主体程序部分:

char cAddr[80], cVal[80];

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
    // user's command interpreter
    // refer to all demo
    int j,iVal,iAddr;
    if (Cmd[0]) // Not Null command
    {
        switch(Cmd[0])
        {
        case 'i': // inp command: i 0 or I0
    }
}
```

```
case 'l':
               sscanf(Cmd+1, "%x", &j);
               iVal=inp(j);
               sprintf(cVal, "%x" ,iVal);
               break:
           case 'o': // outp command: O0 5A or o 0 AB
           case 'O':
               sscanf(Cmd+1,"%x %x", &iAddr, &iVal);
               sprintf(cVal,"%x", iVal);
               outp(iAddr, iVal);
               break:
           default:
               return 0; // return error
       };
       strcpy(Response, cVal); // return I/O value
       return 1;
                 // return OK
   }
   return 0;
              // return ERROR
}
```

## Demo7: 使用函数 printCom1 调试程序

头部说明代码:

// 19~!@#\$ -> Any non-null command will be accepted.

// You can use "Print" or "printCom1" to send Debug string to PC monitor by COM1 of the 7188E.

- // "printCom1" is the defualt function of Xserver.
- // If you want use "Print", you must use "DisableCom" and "RestoreCom"
- // to disable "printCOM1".

// Hardware: [7188E]

主体程序部分代码:

```
int UserCmd(unsigned char *Cmd,unsigned char *Response)
{
    // user's command interpreter
    // refer to all demo
    if (Cmd[0]) // Not Null command
    {
        printCom1("%s\n\r",Cmd); // Send debug string to PC monitor by
7188E's COM1
        strcpy(Response,Cmd);
        return 1; // return OK
    }
    return 0; // return ERROR
}
```

#### Demo9: 计时器触发例程

头部说明代码:

- // 19c, 19C -> clear count value
- // 19r, 19R -> read count value
- // UserCount will be executed every second.
- // Count value will be increased in UserCount.
- // PC can read count value to know number of seconds
- // after count value is cleared.
- // Hardware: 7188E

```
unsigned int cnt;
char c_cnt[20];
void UserCount(void)
{
    // user's timer trigger function
    // refer to demo9 for example code
    // printCom1("\n\rUserCount %d", cnt);
    cnt++;
}
```

```
void UserInit(void)
{
    // user's initial function
   // timer initialized for UserCount()
   // I/O or variables initialized for UserLoopFun()
    // I/O or variables initialized for User's functions in this file
    // refer to demo9 & demo11 for example code
    cnt=0;
    AddUserTimerFunction(UserCount, 1000); // call UserCnt every 1000 ms
}
int UserCmd(unsigned char *Cmd, unsigned char *Response)
{
    // user's command interpreter
    // refer to all demo
    int i:
    if (Cmd[0]) // Not Null command
    {
        switch(Cmd[0])
        {
            case 'c': // 19c, 19C --> clear count
           case 'C':
               cnt=0;
               break;
           case 'r':
                       // 19r, 19R --> read count
            case 'R':
               break;
            default:
               return 0; // Command Error
       }
        sprintf(c_cnt,"%d",cnt);
        strcpy(Response,c_cnt); // return count value
                 // return OK
        return 1;
    }
    return 0;
                            // return ERROR
}
```

## Demo11: 实时 I/O 控制

头部说明代码:

- // 19i, 19I -> read D/I form address 0
  // 19o, 19O -> D/O to address 0
- // This demo show complex real time DI/DO operation in UserLoopFun

// Hardware: [7188EA] or [7188E + D/I/O expansion board] // Note: Base address of 7188EA's DI is 1.

```
unsigned char DI,DO;
char cVal[80];
void UserLoopFun(void)
{
   // VxComm.exe will call this function in every scan time
   // real-time I/O control function
   // DO0=!DI0, DO1=(DI1+DI2), DO2=(DI3)
   // DO3 ~ DO6: control by host-pC
                                     // read DI
   DI=inportb(0);
   if (DI&1) DO=DO&0xfe;
                                     // DO0=!DI0
   else DO=DO | 1;
   if ((DI&2) || (DI&4)) DO=DO | 2;
                                    // DO1=(DI1+DI2)
   else DO=DO & Oxfd:
   if (DI&8) DO=DO | 4;
                                     // DO2=DI3
   else DO=DO & Oxfb;
   outportb(0,DO);
                                     // write to DO0, DO1, DO2
}
```

```
int UserCmd(unsigned char *Cmd, unsigned char *Response)
{
    int iAddr=0;
    char DoVal;
   switch(Cmd[0])
    {
       case 'i':
       case 'l':
           DI=inportb(iAddr);
           sprintf(cVal,"%x",DI);
           break:
       case 'o':
       case 'O':
           sscanf(Cmd+1,"%x",&DoVal);
           sprintf(cVal,"%x",DoVal);
           DoVal=DoVal&0x78;
           DO=DO | DoVal;
           outportb(0,DoVal); break;
       default:
           return 0;
                                       // return Error
   };
   strcpy(Response,cVal);
                                          // return OK
    return 1;
}
```

注意: The key points of this demo are given as follows:

- 主机 PC 可控制硬件 DO3 ~ DO6。
- Xserver 可独立控制硬件 DO0 ~ DO2。
- 以上函数 UserLoop 操作可作为典型紧急事件控制应用。

#### **Demo12**: 扫描周期应用 头部说明代码:

// 19c, 19C -> clear count value
// 19r, 19R -> read count value

- // UserLoopFun will increase count value every scan loop.
- // PC can read how many scan loops after clear count value.
- // So user can use this demo to test the Xserver's performance.
- // Hardware: 7188E

```
unsigned long cnt;
void UserInit(void)
{
    cnt=0;
}
void UserLoopFun(void)
ł
    cnt++;
}
int UserCmd(unsigned char *Cmd, unsigned char *Response)
{
    int i,j;
    if (Cmd[0]) // Not Null command
    {
        switch(Cmd[0])
        {
            case 'c': // "19c" "19C" --> clear
            case 'C':
               cnt=0:
               break:
            case 'r':
                       // "19r" "19R" --> read count
            case 'R':
               break;
            default:
```

```
return 0; // Command Error

}

sprintf(Response,"%d",cnt); // return count value

return 1; // return OK

}

return 0; // return ERROR

}
```

## Demo14: 控制 5 位 7 段 LED

头部说明代码:

```
// 1912345, 19abcde -> Show 5 digits to 7-SEG LED
// Show5DigitLed, Show5DigitLedWithDot can show 5 digits to 7-SEG LEDs
// The two functions can show '0' ~ '9'
// 'a' ~ 'f'
// 'A' ~ 'F'
// ', '-', '.'
// Hardware: 7186E
```

```
主体程序部分代码:
```

#### strcpy(cLED,Cmd);

```
for(i=0;i<5;++i){
    if (cLED[i]>='0' && cLED[i]<='9')
        cLED[i]=cLED[i]-'0';
    if (cLED[i]>='a' && cLED[i]<='f')
        cLED[i]=cLED[i]-'a'+10;
    if (cLED[i]>='A' && cLED[i]<='F')
        cLED[i]=cLED[i]-'A'+10;
    if (cLED[i]==' ')
        cLED[i]=16;
    if (cLED[i]=='-')
        cLED[i]=17;
    if (cLED[i]=='.')
        cLED[i]=18;
    Show5DigitLed(i+1,cLED[i]);
}
strcpy(Response,Cmd);
return 1:
```

#### Demo17: 脉冲宽度测量

头部说明代码:

}

// 19~!@#\$ -> Any non-null command will be accepted.

// This demo is similar to demo13 to detect signal change// and measure signal width.

// Hareware: [7188EA] or [7188EX + D/I expansion board] // Note: Base address of 7188EA's DI is 1.

主体程序部分代码:

unsigned long up\_t; unsigned long down\_t; unsigned high\_width; unsigned low\_width;

// timetick of signal from low becomes high
// timetick of signal form high becomes low
// width of signal stay at high
// width of signal stay at low

```
char now, old;
void UserInit(void)
{
   up_t=0;
   down_t=0;
   if (inportb(0)&1)
       old=1;
                   // initial DI0 is High
   else
       old=0; // initial DI0 is Low
}
void UserLoopFun(void)
ł
   // VxComm.exe will call this function every scan time
   now=(inportb(0) & 1); // read DI0
   if (now != old) // DI0 signal changed
   {
       if (now == 1) // Low become High
       {
           up_t=(*TimeTicks);
           low_width=up_t-down_t;
       }
       else
                      // High become Low
       {
           down_t=(*TimeTicks);
           high width=down t-up t;
       }
       old=now;
   }
}
char cVal[80];
int UserCmd(unsigned char *Cmd, unsigned char *Response)
{
// user's command interpreter
// refer to all demo
   if (Cmd[0])
               // Not Null command
   ł
```

Demo18: 读取 I-7000 系列模块 ID

头部说明代码:

// 19 -> Any command will be accepted.

// This demo shows how to communication with the 7000 series which are connected
// to COM2 of the 7188E.

// Hardware: 7188E + any 7000 series module whose address is 01, // baudrate is 9600.

baudrate is 9600, checksum is disable.

主体程序部分代码:

 $\parallel$ 

```
char cStr[9]; // Receive data from 7000: "!017021" + 0x0c
char cModuleID[5]; // Store module ID: "7021" + 0x00
void UserCount(void)
{
    int i;
    SendCmdTo7000(2,"$01M",0);
    ReceiveResponseFrom7000(2,cStr,1000,0);
    // cut "!01" and cr
    strcpy(cModuleID,cStr+3);
    cModuleID[4]=0;
    // Show moduleID to PC monitor
    printCom(1,"Address: 01, ModuleID:%s\n\r",cModuleID);
}
```

```
void UserInit(void)
{
    AddUserTimerFunction(UserCount, 1000);
    InstallCom(2,9600,8,0);
    SetBaudrate(1,115200L);
}
```

#### Demo19: 读取 64 位全球唯一硬件序列号

头部说明代码:

- // 19 -> Any command will be accepted.
- // Unique hardware serial number is used to protect user's software.
- // Using 7188xw.exe to enter 7188E and then execute command 'ver'.
- // MiniOS7 will show the serial number.
- // User can check the number at first, then decide to execute
- // Xserver forward.

// Hardware: [7188EA] or [7188EX]
// Note: Base address of 7188EA's DI is 1.

```
主体程序部分代码:
```

```
int bSerialNumOk:
char cID[8]={0x9,0x31,0xa4,0x39,0x3,0,0,0x5};
void UserInit(void)
{
   char cSerialNumber[8];
   GetSerialNumber(cSerialNumber);
   if(!strcmp(cSerialNumber,cID))
       bSerialNumOk=1;
                                     // Matching !
   else
                                     // Unmatched !
       bSerialNumOk=0;
}
int UserCmd(unsigned char *Cmd, unsigned char *Response)
ł
   if(bSerialNumOk)
       strcpy(Response, "ID ok.");
```

```
else
strcpy(Response,"ID error.");
return 1: // return OK
```

```
}
```

Demo23: 返回所有数据(包含 "23")到指定客户端

头部说明代码:

- // 23 -> Any command will be accepted.
- // This demo illustrates how to use command 23 to send
- // messages to specific client.
- // Xserver will echo all data (inclued "23") to client.

// Hardware: 7186E

```
int VcomUserBinaryCmd(TCPREADDATA *p)
{
    /* VXCOMM.EXE 2.6.12(09/04/2001) or later will support this function.
        TCP PORT 10000, command 23 will call this function.
        user can get the following message:
        p->ReadUartChar : the buffer store the command data(include "23")
        p->Length : the command data length(include the two byte "23")
        p->Socket : the socket number that receive the command, that is
    when the user function
    want return message to the client, just use the socket to send data.
        use: VcomSendSocket(p->Socket,pdata,datalength);
    */
    // echoes all data (include "23") to specific client.
        VcomSendSocket(p->Socket,p->ReadUartChar,p->Length);
    return 1; /* any value will be accepted */
```

# 5.术语表

## 1. Ethernet(以太网):

现行的以太网通常引用由 Digital Equipment 公司、Inter 公司和 Xerox 公司于 1982 年联合发布的标准构建。在局域网技术大行其道的今天,基于 CSMA/CD 技术开发的收发系统的以太网最为有效,而此主 机通常由 48 位 MAC 地址构成。

#### 2. Internet(因特网):

总体上,报文分组交换网络的采集工作,可通过 TCP/IP 协议进行互 连来完成,且该网络可在单一、大型及一些虚拟网络上运行。另外, 主机因特网通常使用 32 位 IP 地址进行区分。

#### 3. TCP/IP:

传输控制协议 (TCP) 及网际协议 (IP)均为标准的网络通讯协议,它们 在实际应用中通常同时执行,因此常常称其为 TCP/IP 协议。TCP/IP 协议可大量应用于大量交互网络的通讯工作。

#### 4. TCP (传输控制协议):

**TCP** 协议在两台主机中提供可靠的数据流服务。基于该协议的连接通常伴随有应用程序中以适当大小进行封包的数据在网络层中进行传输,而这些数据封包常常包含有设定的超时时长,用于到时接收目标封包来确定另一个封包是否成功的传输,如此往复。

## 5. UDP (用户数据报协议):

UDP 协议在应用层提供一种更为简易的服务。其仅仅只需从主机向外 发送数据封包,而不担保该信息是否成功到达目标主机。

## 6. Gateway(网关):

连接两个网络并进行封包的传递的计算机可称为互连网关或互连路 由。而网关传递封包的目标并非是目标主机,而是基于目标网络。

#### 7. IP (网际协议) 地址:

每个在因特网上的接口或平台必须拥有一个全球唯一的 IP 地址 (也可称为 Internet 地址)。这些地址由分为 4 段十进制数表示的 32 位数字组成,例如: "192.168.41.1"。该数字也可称为点分十进制符号。

#### 8. MAC (媒体存取控制) 地址:

为了让计算机可以判定封包是否可以接收,每台计算机在以太网上都标识有一个 48 位整数,该数字即为 MAC 地址 (通常也可称其为以太网地址、硬件地址或物理地址)。而这些数常常使用 8 位 16 进制数来表示,如:"00:71:88:af:12:3e:0f:01"。所有以太网硬件制造商必须向指定的国际统一标准协会购买这些 MAC 地址段,在硬件制造中依次使用不同的 MAC 地址,这样就可以确定世界上没有任何两个以太网硬件接口拥有同相的 MAC 地址。

#### 9. Subnet Mask(子网掩码):

子网掩码常被简称为掩码。 Given its own IP address and its subnet mask, a host can determine if a TCP/IP packet is destined for a host that is (1) on its own subnet, (2) on a different network. If (1), the packet will be delivered directly; else, will be delivered by gateways or routers.

#### 10.ARP (地址解析协议):

考虑到如果有两台设备 A 和 B, 共享着一个网络。每台设备都有各自的 IP 地址(如: IP<sub>A</sub> 和 IP<sub>B</sub>)及 MAC 地址(如: MAC<sub>A</sub>和 MAC<sub>B</sub>)。需要这样一个基于 MAC 地址的底层软件来处理如何让应用程序仅仅只通过 IP 地址就可以完成两者的通讯应用工作。但从根本上, 两者若要完成 通讯, 在硬件层上无论如何都必须经由 MAC 地址分类来执行相关的 工作。

倘若设备 A 与和其已经连接的设备 B 之间需要发送数据封包,而仅仅 只知道设备 A 和 B 的 IP 地址。这样问题就产生了:两者之间的 MAC 地址,如何进行映射?

ARP 提供一种从 32 位 IP 地址到 48 位 MAC 地址的动态映射通讯。这样,我们就可以通过该协议动态的自动处理相关事宜,程序应用人员或系统管理员就可不再关心这些地址。

11. RARP (反地址解析协议):

RARP 提供从 48 位 MAC 地址到 32 位 IP 地址的反向动态映射通讯。



#### 12.ICMP (网际控制报文协议):

世上没有任何系统可以永久无故障地运行。ICMP 协议可以为两个基于网际协议软件(IP)提供通讯服务。该协议允许风头发送错误或控制指使给另一个网关或主机,以告之网络是否发生通讯故障。

#### 13. Ping:

指令 Ping 为获得一个有效响应,需要发送一个 ICMP 返回请求信息给 目标主机。通常情况下,若主机 Ping 不通,则将不能与主机进行 Telnet 或 FTP 服务。换句话说,若与主机不能正常进行 Telnet 或 FTP 这样的服务,指令 Ping 常常可作为解决问题的出发点,优先进 行。

#### 14. Packet(封包):

所有数据单元均可通过网络进行传输,这样的数据单元即为封包。它 通常包含有关键数据、控制信息、源节点及目标(如:主机)节点地 址,且以固定的报文格式进行传输

#### 15. Socket:

每个 TCP 信息段包含有源码及目标端口号用于发送及接收应用数据的 识别。这两者数据与源码及目标 IP 地址均放于 IP 报头,以特有方式 区别每一个连接。

IP 地址及端口号的组合即为 socket。

#### 16. Clients and Servers(客户端与服务器):

客户端与服务器例程初始阶段都要分区程序是客户端还是服务器。总体说来,一个应用程序通常以点对点方式进行初始化通讯,因此可以 理解为客户端。当用户使用网络服务时,就常常调用这些客户端程 序。

大多数客户端程序包含有应用程序开发工具。每次客户端程序执行 时,都要发送请求命令并等待响应,当响应到达时,才可继续正常工 作。此外,客户端程序通常情况下较服务器端更容易开发,也没有特 定的操作系统要求顾虑。

比较起来,服务器端程序需要等待来自客户端的进入请求指令,并且 也只有当接收到客户端的请求指令,它才可执行后续的逻辑运算处理 功能并将结果返回给客户端。

#### 17.Firmware(固件):

所有程序均可储存在永久性存储器中,诸如: ROM, EEPROM 或 Flash 记忆体。