

## 基于 WinCE 的大容量程控数据传输解决方案设计

任云, 师奕兵, 付在明

(电子科技大学 自动化工程学院, 四川 成都 611731)

**摘要:**给出了一种基于 Client/Server 网络通信模型, 以 Socket 为编程接口, 嵌入式测量设备为控制目标的远程控制软件设计方案。详细介绍了网络通信中的 TCP/UDP 协议、非阻塞方式、select I/O 模型以及多线程同步等关键技术。最后, 通过高性能脉冲/数据发生仪的远程控制软件应用实例验证了该方案的有效性。

**关键词:** Windows CE. NET; 远程控制; 套接字; 非阻塞; 多线程同步

**中图分类号:** TP393    **文献标识码:** A    **文章编号:** 1000-8829(2009)12-0084-04

### New Programme Design of High-Capacity Data Transmission Based on WinCE System

REN Yun, SHI Yi-bing, FU Zai-ming

(School of Automation, University of Electronic Science and Technology of China, Chengdu 611731, China)

**Abstract:** A design of remote-control software targeted at embedded measuring instrument, which is based on the model of Client/Server network communication with Socket is represented. Some key technologies over network communication field are discussed in detail, such as TCP/UDP protocol, non-blocking, select I/O pattern and multi-thread synchronization. A practical application of remote-control software for pulse/data generator is provided to verify the feasibility and validity of this design.

**Key words:** Windows CE. NET; remote-control; Socket; non-blocking; multi-thread synchronization

远程控制是指由一台设备通过远程控制软件实现对另一台设备的远距离操作和控制, 在某种程度上颠覆了传统仪器在时间和空间上的限制, 为用户提供了一种不必亲临现场的便捷操作方式, 节约了时间与成本。随着测量技术的发展, 许多高速高精度的测量设备也要求具备远程控制功能, 它们对于实时性、大容量数据传输以及网络通信的稳定性有着更高的要求。

Windows CE. NET 是一个多任务、可定制的嵌入式实时操作系统, 满足便携式测量仪器、信号发生器等精密测量设备实时、高效的要求, 在测量系统中被广泛采用。在测试测量领域, 提供分立式仪器与计算机之间互连性的称为外部总线, 包括 GPIB 总线、USB、LAN 等。GPIB 总线传输速率只有 8 Mb/s; USB 排线没有工业标准规格, 且两者的传输距离均不超过 30 m; 而 LAN 总线以其速度快、低成本、传输距离远的特性更适于测试系统的远程控制。

本设计给出了一种 Windows CE 嵌入式测量仪器的远程控制软件设计方案, 采用 LAN 总线, 选用 Socket 编程接口, 并结合非阻塞、多线程同步等先进技术, 针对其大容量数据传输的问题提出了解决办法。

### 1 远程控制软件功能与结构

基于 C/S (Client/Server) 模式开发的远程控制软件包括 3 个组成部分: 服务器、客户端和网络传输单元 (如图 1 所示)。嵌入式测量仪器作为服务器, 运行在 Windows CE. NET 平台下; 客户端是一台或多台计算机, 运行在 Windows 操作系统下。

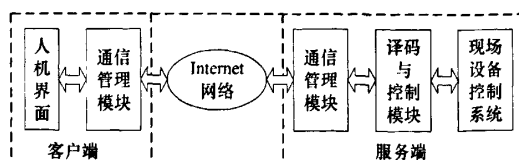


图 1 基于 C/S 模式的远程控制

客户端包括人机界面和通信管理模块。人机界面为用户提供友好交流的平台, 用户通过人机界面下达

收稿日期: 2009-08-11

作者简介: 任云(1984—), 女, 在读硕士研究生, 主要研究方向为电子测量仪器、高性能脉冲/数据发生仪等。

控制命令;通信管理模块负责与服务端的通信管理。

服务端分为通信管理模块、译码与控制模块以及现场设备控制模块。通信管理模块负责与客户端的通信管理,向用户反馈连接状态,接收客户端数据包;译码与控制模块负责解析数据包、调用本地函数向现场设备发送控制字,达到控制输出信号的目的<sup>[1]</sup>。

Internet 网络是服务器和客户端之间远距离传输指令和信息的纽带。开发测量设备的远程控制功能时,鉴于其高速高精度以及大容量的特性,首先要保证数据传输的可靠性、安全性及实时性,故采用 TCP/IP 协议通信。

## 2 远程控制软件设计

### 2.1 用户界面设计

客户端用户界面的开发采用 Visual C++ 6.0,服务端界面开发采用 eMbedded Visual C++ 4.0,两者均为用户提供了一些常用的标准程序结构和编程风格,能满足远程控制用户界面的设计需求。

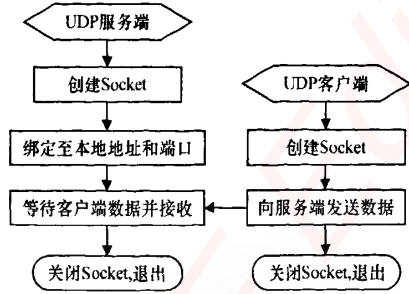
### 2.2 通信模块设计

#### 2.2.1 Client/Server 通信模式

服务器与客户端的通信采用 Socket 编程接口,在 ISO 的 OSI 网络 7 层协议中,Socket 屏蔽了数据链路层和物理层,主要负责传输层和网络层,使编程简单。C/S 模式的 Socket 编程有 2 种:基于 TCP(面向连接)和基于 UDP(面向无连接)<sup>[2]</sup>,两者远程控制的流程如图 2 所示。

在嵌入式测量仪器的远程控制设计中,一般要求可靠链路传输与实时传输,且丢包率小,因此 TCP 更能满足数据传输的要求,故在此选用面向连接(TCP)的 Socket 编程。

基于 TCP 的网络通路建立过程如下:服务端首先创建一个 Socket 套接字,调用 bind() 函数绑定至本地



(b) 基于 UDP 的远程控制流程图

图 2 基于 TCP/UDP 的远程控制流程图

地址和某一个端口,然后调用 listen() 函数将 Socket 设置为监听模式,等待客户端的连接请求;客户端创建一个套接字后,调用 connect() 函数向服务端发送连接请求;服务端调用 accept() 函数接受客户端请求并返回一个新的会话套接字,新的会话套接字负责与客户端通信。

建立网络通路后,用户在客户端人机界面设置各项控制参数,依据数据传输协议对控制参数编码,然后调用 send() 函数经由 Internet 网络发送至服务器端的网络通信模块;服务端接收控制信息、解码,并根据控制参数向底层硬件发送相应控制字,最终可在示波器上观测到信号输出。

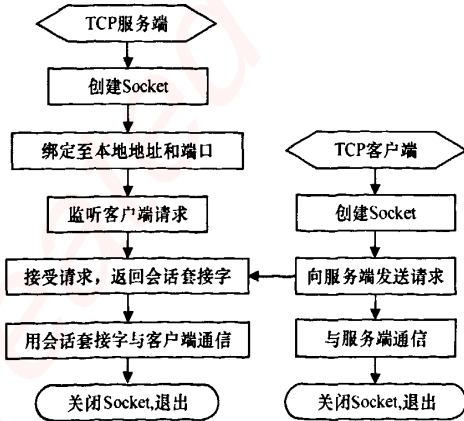
#### 2.2.2 非阻塞模型

套接字有两种工作模式:阻塞和非阻塞,套接字默认工作方式是阻塞。基于 TCP 的 Socket 编程中,accept() 函数、recv() 函数都是阻塞式的。服务端调用 accept() 函数接受连接请求时,若套接字上未检测到连接请求,accept() 会一直阻塞直到有连接请求出现;同样,recv() 函数执行接收操作时也会一直阻塞直至会话套接字上有数据可读,并把数据读到接收缓存区后 recv() 函数才会返回。在单线程的程序里出现这种情况将会导致主线程阻塞,从而整个程序被锁死<sup>[3]</sup>。

为避免出现这种问题,可调用 ioctlsocket() 函数将套接字设置为非阻塞模式,非阻塞方式的套接字执行函数后会立即返回,至于操作是否执行成功可通过 select I/O 模型检测。select() 函数的原型是:

```
int select (int nfd, fd_set FAR * readfds, fd_set FAR * writefds, fd_set FAR * exceptfds, const struct timeval FAR * timeout);
```

第 1 个参数被 Windows 系统忽略,第 2 个参数用来检查套接字的可读性,对于一个处于监听状态的服务端套接字,等待 accept() 函数处理的 connect() 连接请求也被视为此套接字的可读数据;第 3 个参数检查可写性,第 4 个参数检查意外数据,最后一个参数是一个时间结构体,表示 select() 函数的最长等待时间,超



(a) 基于 TCP 的远程控制流程图

过等待时间立即返回。select() 函数返回值大于 0 表示套接字有待处理的请求,用 FD\_ISSET 检查套接字句柄,如果套接字句柄仍然在开始分配的 fd\_set 里,说明应该立即处理<sup>[4]</sup>。

### 2.2.3 多线程以及线程间同步

高性能嵌入式测量设备的远程控制功能需要实现一台嵌入式设备与多台计算机的远程通信,并且发送/接收数据量非常大,而在主线程中发送/接收大容量的数据会影响用户的界面操作,降低应用程序工作效率。在高性能嵌入式测量设备远程控制的数据传输设计中,为了不影响用户界面操作,将大容量程控数据的传输放到独立的子线程中处理,每个线程轮流占用 CPU 的运行时间和资源。由于 CPU 的处理速度非常快,且每个时间片又很小,给人的感觉是多个线程在同时运行。创建子线程用 CreateThread() 函数,由于 Windows CE.NET 系统不支持安全性,故第 1 个参数必须设置为 0。

多线程设计思路为:服务端创建两个子线程,即负责处理客户端连接请求的子线程 A 和负责接收客户端数据的子线程 B,子线程中均采用 select I/O 机制检测非阻塞套接字。在子线程 A 中循环检测已处于监听状态的套接字的可读性,有连接请求则调用 accept() 函数接受客户端请求同时返回新的会话套接字。而旧的套接字则继续利用 select 机制检测其他客户端请求,实现了多台计算机与一台服务器的远程连接。在子线程 B 中循环检测线程 A 中创建的会话套接字的可读性,有数据则执行接收操作,否则继续检测。CPU 通过在特定的时间片内轮询各个线程,提高了工作效率,同时避免了在主线程中循环检测引起主程序阻塞。

子线程 B 访问子线程 A 中创建的会话套接字。若一直未检测到客户端请求,则会话套接字不存在,此时子线程 B 访问不存在的会话套接字将会导致数据冲突,引起数据混乱,这就是多线程中常涉及的线程冲突问题,解决办法是实现线程间的同步。

Windows CE 操作系统提供了 Event(事件)、Mutex(互斥对象)、Semaphores(信号量)和 Critical-section(临界区)4 种同步控制对象来实现线程间的同步。事件对象是实现线程同步最常见的方法之一,当线程访问某一资源前需要等待某一事件发生时,最适合使用事件对象。事件对象可以处于有信号或者无信号两种状态<sup>[5]</sup>。

用 CreateEvent() 函数创建一初始状态为无信号的事件对象 m\_Event,子线程 A 中调用 accept() 函数成功创建会话套接字后立即置 m\_Event 为有信号状态。子线程 B 中监视 m\_Event 状态,若 m\_Event 处于

无信号状态,则需等待其变为有信号状态;当 m\_Event 处于有信号状态时,表示可以执行同步操作,线程 B 可以监测会话套接字,以便在适当的时候执行读操作。等待函数的原型是:

```
DWORD WaitForSingleObject( HANDLE hHandle, DWORD dwMilliseconds );
```

如果第 2 个参数为 INFINITE,且 m\_Event 永远不变为有信号状态,那么调用线程永远不会被唤醒,永远处于死锁状态,不过这不会浪费宝贵的 CPU 时间<sup>[6]</sup>。

## 3 试验结果

将嵌入式测量设备的远程控制软件设计方案应用于高性能脉冲/数据发生仪的远程控制软件开发中,鉴于脉冲/数据发生仪参数众多、数据精度要求高、传输数据量大<sup>[7]</sup>,而 send()/recv() 函数发送/接收数据缓存区参数类型是 char\*, 因此非数值型参数传输时用 char 变量的连续两个二进制位表示,以工作模式为例,00 表示连续模式,01 表示猝发模式,10 表示门控模式。这样一个 char 字符便可表示 4 个非数值型参数,极大地压缩了传输数据包的大小,缓解了高性能脉冲/数据发生仪大容量数据传输引起的传输阻塞和费时间问题,提高了工作效率。试验结果如下。

### 3.1 脉冲/数据发生仪远程控制的部分代码实现

以 recv() 函数为例,子线程中实现非阻塞与线程同步的部分代码如下:

```
while (TRUE) {  
    //等到接受客户端请求后置 m_Event 为有信号状态,实现了线程同步  
    if (WaitForSingleObject( pDlg -> m_Event, INFINITE) == WAIT_OBJECT_0) {  
        //检测读事件  
        ret = select(0, &fdRead, NULL, NULL, &aTime);  
        if (ret > 0) {  
            //判断是否所关注套接字上的读事件  
            if (FD_ISSET( pDlg -> m_socket, &fdRead)) {  
                recvLen = recv( pSocket -> m_socket, recvBuf, len, 0); //执行接收操作  
                ...  
                //处理数据}}}}}
```

### 3.2 建立脉冲/数据发生仪远程控制的网络通路

使高性能脉冲/数据发生仪处于监听状态,在客户端人机界面上输入脉冲/数据发生仪的 IP 地址和监听 Port,发送连接请求;脉冲/数据发生仪用户界面实时刷新显示已连接的客户端 IP 地址,如图 3 所示。

### 3.3 脉冲/数据发生仪远程控制现场设备输出测试

在客户端人机界面的“模式/触发”页面设置各项参数如下(如图 4)。

与脉冲/数据发生器建立连接的客户端列表

IP地址	设备名	连接状态
218.194.59.175	Administrator	已连接
218.194.59.175	Administrator	已断开
218.194.59.149	Trancy	已连接

图3 已连接客户端列表视图

① 连续模式, 信号类型为群脉冲, 频率为 40 MHz, 群脉冲周期为 6CLK;

② 通道 1 群脉冲个数为 2, 延迟为 10 ns, 脉宽为 8 ns, 高电平为 1.2 V、低电平为 0.0 V;

③ 通道 2 群脉冲个数为 5, 延迟为 0 ps, 脉宽为 12.50 ns, 高电平为 1.0 V、低电平为 0.0 V。

发送远程控制命令, 脉冲/数据发生器接收程控命令并控制信号输出, 输出信号波形通过示波器显示如图 5 所示(图中由上至下, 依次是触发信号、通道 1 输出、通道 2 输出波形)。



图4 “模式/触发”页面



图5 40 MHz 群脉冲的远程控制输出波形

节指令只有操作码,隐含操作数;双字节指令,第1个字节是操作码,第2个字节是操作数;三字节指令第1个字节是操作码,后两个字节是操作数。CPU在取指令的时候是先取操作码,再取操作数。如何判断是操作码还是操作数就是通信取指令的顺序,而取指令的顺序完全由指令计数器PC来控制。当单片机系统受干扰出现错误,程序便会脱离正常轨道,出现“乱飞”。当“乱飞”到某双字节指令,若取指令时刻落在操作数上,误将操作数当作操作码,程序将出错;若“乱飞”到三字节指令,出错几率更大。这样就要在一些对程序的流向起决定作用的关键指令前人为地插入一些单字节指令,或将有效单字节指令重写,此即冗余指令<sup>[6]</sup>。

本系统下位机软件设计中采取了在双字节指令和三字节指令后插入两个字节以上NOP指令的措施,这样即使“乱飞”程序飞到操作数上,由于空操作指令NOP的存在,避免了后面指令被当作操作数执行,程序自动纳入正轨。此外,还在对系统流向起重要作用的指令如RET、CALL等指令之前插入2条或3条NOP指令,也可以将“乱飞”程序纳入正轨,确保这些重要指令的执行。

### 4.3 软件陷阱技术

89C52单片机内部存储器中除了程序区外,还有非程序区,这样,当“乱飞”的程序进入非程序区的时候,可设定软件陷阱对“乱飞”程序进行拦截,从而将程序引向一个固定的位置,即可将捕获的程序重新纳入正轨。软件陷阱主要就是程序重新引入它的复位入口处,也就是说在程序适当的地方设置这样一个指令:

```
NOP NOP LJMP 0000H.
```

(上接第87页)

## 4 试验结论

以上给出了一种可行的嵌入式测量设备的远程控制软件实现方案,在不改变嵌入式测量设备本地控制系统的基础上,在一个局域网中,通过给脉冲/数据发生仪指定的唯一IP地址和端口,实现与网内任意一台计算机的远程连接。

结合高性能脉冲/数据发生仪的应用实例,采用非阻塞、select I/O机制、多线程以及多线程同步等技术,解决了大容量数据网络传输中的阻塞与低效率问题,提高了数据传输的工作效率和稳定性。

### 参考文献:

[1] 潘丹青,田书林,刘科.网络故障测试仪远程监控软件的设计[J].测控技术,2006,25(10):65-67.

### 4.4 软件“看门狗”技术

89C52单片机中有3个定时器,其中两个定时器T0和T1来对程序的运行进行监控。这样就构成了一个循环,T0监视T1,T1监视主程序,主程序又来监视T0,从而保证系统的稳定运行。

## 5 结束语

本设计针对小型活塞式无人机发动机试验检测系统的应用要求,综合运用虚拟仪器、串行模数转换、软件抗干扰等成熟技术,实现了对发动机遥控指令产生、遥测参数采集、测试数据智能分析、实时故障提示和维修资源共享等具体功能,体现了军用装备智能检测技术的发展方向。研究及试验结果表明:系统的可靠性和测试准确度完全达到了设计要求,检测参数全面,人机界面友好,使用维护简单方便,提高了无人机发动机系统野战条件下的试车效率。

### 参考文献:

- [1] 张加圣,应英,应勇.航空发动机故障监控系统的设计与软件实现[J].测控技术,2008,27(2):65-68.
- [2] 李永科,陈自力,田庆民.无人机发动机点火停车电路故障在线检测研究[J].测试技术学报,2003,17(3).
- [3] 王英,沙云东.航空发动机故障诊断技术综述[J].沈阳航空工业学院学报,2007,24(2):11-14.
- [4] 杜建红,乔黎.航空发动机试车台自动测试系统设计[J].航空发动机,2007,33(2):15-17.
- [5] 杨乐平,李海涛,杨磊.LabVIEW程序设计与应用[M].北京:电子工业出版社,2005.
- [6] 李朝青.PC机及单片机数据通信技术[M].北京:北京航空航天大学出版社,2001-01.

- [2] 孙鑫,余安萍.VC++深入详解[M].北京:电子工业出版社,2006:531-532.
- [3] 汪兵,李存斌,陈鹏.EVC高级编程及其应用开发[M].北京:中国水利水电出版社,2005:329-332.
- [4] 丁展,刘海英,等.Visual C++网络通信编程实用案例精选[M].北京:人民邮电出版社,2004:1-26.
- [5] Boling D.Windows CE程序设计[M].北京:北京大学出版社,1999.
- [6] Nick G,Marshall B.Windows CE 3.0 application programming[M].New Jersey,Prentice Hall PTR,2000.
- [7] Agilent Technologies Copyright.Agilent 81130A 400/660MHz pulse/data generator quick start guide[EB/OL].<http://cp.literature.agilent.com/litweb/pdf/81130-91021.pdf>,2001-05-01.

# 嵌入式资源免费下载

## 总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)

## VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)

## Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)

## 6. [嵌入式 C 进阶之道](#)

## Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)

## PowerPC:

1. [Freescale MPC8536 开发板原理图](#)

## ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)

## Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)