文章编号：2095-4980(2018)03-0535-06

协议级综合调试器设计

蔡文斋

(中国电子科技集团公司 第三十九研究所，陕西 西安 710065)

摘　要：为解决工程软件中各种接口调试问题，设计了一款综合调试器。该调试器使用多种接口方式与被调程序通信，硬件接口支持串行口通信、网络用户数据报协议(UDP)、传输控制协议(TCP)、网络组播方式；软件接口使用文件方式、WINDOWS WM_COPYDATA 消息通信方式、剪切板进程通信方式、文件映像通信方式。设计了一种特殊协议定制方法，任何复杂的接口协议均可定制性发送，这样被调试软件可得到需要的数据；同时设计了分析功能，可以监视被调软件发出的协议数据的正确性。该调试器在任何一种通信方式下，均支持到位级的数据编辑功能。使用该综合调试器可使开发者在实验室内快速开发出全套监控应用软件，也可在综合测试中快速分离出双方的通信问题。

关键词：调试器；进程通信；串行口；网络通信；网络通信组件

中图分类号：TN345　　**文献标志码：**A　　doi：10.11805/TKYDA201803.0535

Protocol level integrated debugger design

CAI Wenzhai

(The 39th Research Institute of China Electronics Technology Group Corporation, Xi'an Shaanxi 710065, China)

Abstract: An integrated debugger is designed for resolving a variety of monitoring program interface debugging problems in engineering application. The debugger utilizes multiple interface methods with modulated program communication. The hardware interface support serial port communication, network, User Datagram Protocol(UDP) and Transmission Control Protocol(TCP), multicast way. Software interface utilizes the file, WINDOWS WM_COPYDATA communication, clipboard process communication, file map communication. Special protocol customization method is designed, by which any complex interface protocol can be customized to send, therefore the debugging software can get the required data. The analysis function is designed to monitor the correctness of the protocol data issued by the software. In any form of communication, a level of data editing function can be supported. This integrated debugger allows developers to quickly develop full set of monitoring applications in the laboratory, and to quickly extract communication problems from both sides.

Keywords: debugger; process communication; serial port; network communication; internet direct

在航天测控、工业控制、多种科研专题研究项目中，承研者的计算机可能连接了各种传感器或者计算机，在开发项目软件时如果在实验室可提前模仿出各种外接接口的通信情况，则可以在联调时节约大量时间。项目中通信协议是五花八门的，是否可以设计出一款灵活的协议调试器，通过软件界面定制这些形式各异的通信协议，解决各种变化协议的调试问题，本文调试器正是基于这样一种需求而设计。作者承研过多种航天测控项目(天线类，光学类，天文类，仿真类等等)，光学、天文类测控项目中使用了多种形式的板卡，在开发中经常碰到的首要问题就是通信调试问题，作者曾写过各种接口的调试器以供工程使用，这次设计了协议级的调试机制，将多种调试方式集中在一起，最终形成了这款通用的协议级调试器[1]。

1　总体需求

假定某工程要求承研者设计如下监控系统(8 路串行口设备，3 路网络设备，64 路 I/O 设备，8 路 A/D 设备，

收稿日期：2016-10-19；修回日期：2017-02-16

3 路 D/A 设备)。形式化的接口如图 1 所示。

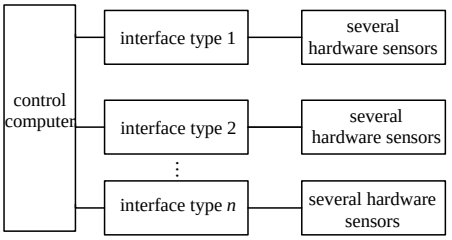

Fig.1 Control system hardware interface
图 1 控制系统硬件接口示意图

从软件角度考虑,控制项目中可能有 m 个类型的接口形式(例如:串行口、网络接口、A/D 接口、D/A 接口、I/O 接口、CAN 接口等),如图 2 所示。每个类型下有若干个传感器,每个传感器与计算机通信可能有 2 种软件通信协议,一个用于读,一个用于写,形式化的通信结构应为 A:StructureWrite,StructureRead。在结构 A 中,通信双方实际使用的仅仅是一个抽象的缓冲区,程序员使用时,将依据通信协议写为一个确定的结构形式。

考虑某一硬件传感器与计算机的通信情况,假定该传感器写给计算机的结构为 B(C 语言形式)。

```
StructureWrite   {BYTE    id,
                  BYTE    Status[6]
                  Int     INT10[10]
                  Float   float0[8]
                  Double double9[9]
                  }
```

同理,该传感器从控制计算机读的数据结构为 StructureRead,这个读结构可能与写结构是不同的,当然是另一组元素包。

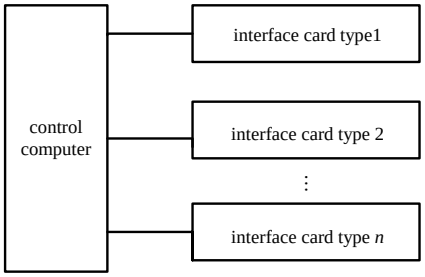

Fig.2 Hardware interface
图 2 硬件接口示意图

任一硬件传感器(包括某计算机)要和另一台控制计算机通信,读写结构基本上是不同的,在实验室内,当不连接外置具体传感器时,如果已知通信协议,使用软件方法帮助开发者快速调通应用程序,将会为工程项目研发在联调阶段节约许多时间。如果能设计一款调试器,定制化地产生这些通信协议结构内元素数据(注意传感器同计算机通信协议是千变万化的),并且可以按照硬件接口方式传出协议数据,那么应用程序就应在实验室内完全调通。

工程用计算机常用的硬件接口为串口和网络,一般在购置计算机时是标配,其他硬件接口需要购买硬件相关板卡。

无论传感器是什么接口或者是某计算机,承研者最关心的问题是怎样得到按照通信协议的一组数据结构内元素数据,只要有某种手段将这样一种结构的内容按照通信协议构造出来,并使得被调试应用程序能够接收到,那么在实验室内就可以完成监控软件开发与调试,一旦各传感器购置到位,并配置好驱动程序,那么监控软件的开发就基本完成。

综合调试器设计目标应为:

1) 调试器应具备常用通信接口、十六进制编辑输入功能、十六进制显示功能、位级的编辑功能、定时发送功能、任何通信协议的定制功能以及多种通信接口;

2) 应支持任一其他类型硬件接口仿真调试;

3) 应分析出由工程应用程序发出的按照某协议传出的数据的正确性。

2 调试器调试原理

任何硬件传感器或者计算机同另一台计算机通信协议只有 3 种方式:第 1 种为纯 ASCII 码字(纯字符型);第 2 种为 ASCII 加特殊字符型;第 3 种为某结构型数据包。第 1 和第 2 种形式的协议易于实现,这种通信协议常见于国外的各种传感器,第 3 种形式的协议可能五花八门,工程项目不同,形式各异。由调试器产生出任何工程中所需要的任何形式的数据,并通过某种通信方式传给应用程序,这就是调试的写;当控制计算机需要分析从传感器读入的数据时,如果调试软件能够显示出读入的原码字,并且能够显示出按照通信协议约定的通信结构内每个元素值,这才是真正的读调试本质。结构化数据的每个元素值,通过硬件或者软件方式传给应用软件,那么应用工程软件就相当于读到了某硬件传感器发送的数据(注意调试器可产生按通信协议规定的任何数据),这样,承研者开发的应用软件即可在实验室完成绝大部分工作。那么控制程序中控制部分调试也就完成了。同理,应用程序发给调试器的数据结构如果也可以由调试器分析出每个结构内数值,那么控制程序中监视部分调试也就完成了。调试器首先应具备 2 大功能:产生任何结构内元素数据;分析出任何协议结构内数据。

应用监控计算机所连接的所有这些外置设备视为某传感器,不妨视为某一抽象传感器,应用程序同这些传感

器的通信问题是按某通信协议的一串码流，视为 2 个缓冲区，读与写，应用软件同这些传感器的通信实质实际是一串 2 进制的码流传输，使用软件方式书写可以写为形式化的结构，暂称为结构 A。使用 Pascal 语言写为如下形式：

```
TYPE Sensor Structure
SensorId:BYTE;//传感器号
ReadBuffer:Tbytes;//读缓冲区
WriteBuffer:Tbytes;//写缓冲区
ReadLength:integer;//读长度
WriteLength:integer//写长度
End
```

应用程序中使用的数据实际涉及到 2 个数据结构：通信用的结构 A；通信协议结构 B。如果使用内存一字节对齐技术，那么可以使用 MemoryCopy 函数完成数据结构内容互换。内存一字节对齐技术在不同程序语言中表达方式是不同的，在数据结构中声明 Delphi 使用 packed 关键字，C 语言使用条件编译#pragma pack(push)//保存内存对齐状态。

```
#pragma pack(1)
结构声明
#pragma pack(pop)   //恢复内存对齐状态。
```

协议结构内部的元素无论使用什么操作系统和编程语言，核心的数据都只有下面几种类型：BOOL,CHAR,BYTE,WORD,DWORD,INTEGER,SINGLE(单精度浮点数),DOUBLE(双精度浮点数)，所谓的通信协议是将这些类型的数据按照某种方法排在一起，排法可以有无穷多种，所以工程中使用的通信协调是五花八门的。协议结构中最复杂的数据表示是浮点数，浮点数一般使用内存表达或者使用整型量补码表示，有些通信系统为了节约使用字节长度，可能使用 2~3 个字节表示浮点数，例如使用 2 个字节表示 0~360 之间的数据，协议定制中也设计了该部分定制功能，这样就解决了所有复杂类型的定制问题。

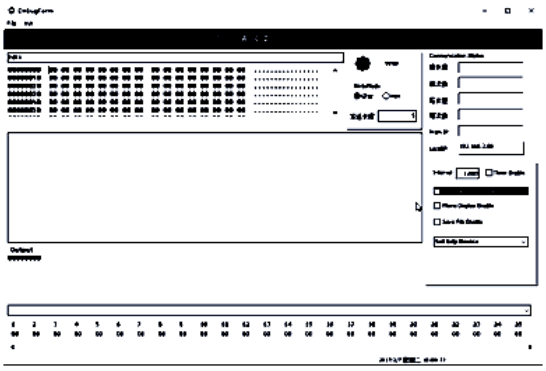

Fig.3 Debugger interface effect
图 3 调试器界面效果截图

3　总体设计

调试器总界面中给出十六进制的输入输出部分，通信的各种状态信息部分组合的网络配置界面如图 3 所示。其中网络初始化部分设计如图 4 所示。

首先应设计出调试器常用的通信部分，硬件级接口支持网路通信与串口，软件级通信应具备多种进程通信技术，将产生的数据传给应用工程软件。

硬件接口设计为串口形式与网络形式，其中网络通信设计出常用的通信协议支持(UDP 方式、TCP/IP 客户端、TCP/IP 服务器、组播)。

软件接口设计为进程通信方式，分别使用文件方式、WU-COPYDATA 消息方式、剪切板方式和文件映像方式。

通信设计：

调试器使用 Rad Studio Delphi XE10 设计；硬件通信部分之串口使用 COMPORT 组件设计；网络部分使用 XE10 的 Indy10 组件实现，Indy(internet direct)网络组件库由国际开源组织开发，全套组件有数百个，已经内置在 Delphi 开发环境内，该调试器分别使用 UDPClint,UDPServert、TCPClint,TCPServert 组播组件等设计[2]。Indy10 下的网络

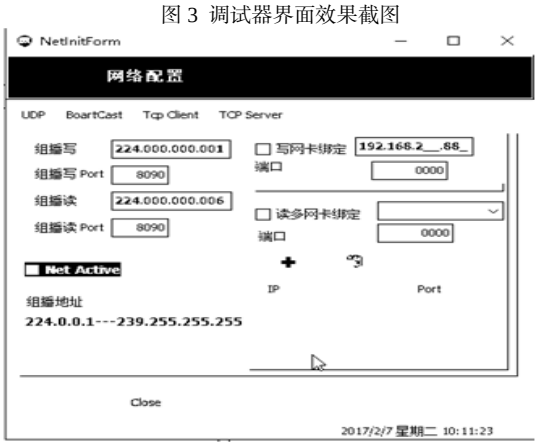

Fig.4 Network initialization interface
图 4 网络初始化界面图

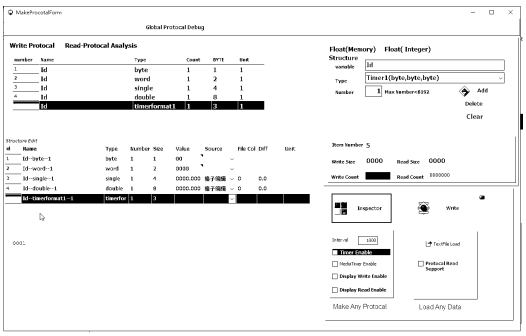

Fig.5 Protocol debugging interface
图 5 协议调试界面截图

编程与 Indy9 大不相同,Indy10 下网络读写已经改为线程级，作者将该部分通信全写为函数形式，函数中的网络组件参数将程序代码段与界面完全分离[3]。

软件方式通信部分是为其他硬件接口调试而设计的(一般情况下工程用计算机的硬件板卡只有在购置后才能使用)，但该调试器无需连接硬件板卡，同样可以产生出该硬件传感器通信的所有数据。

由调试器产生出协议级数据，通过进程通信方式传给应用监控软件，相当于任一其他硬件接口形式的传感器与应用程序通信部分可以进行调试[4]。

4 协议调试器设计

协议调试器由主界面的菜单呼出，界面如图 5 所示(读与写设计为不同的定制功能)。功能区主要由 3 大部分组成：协议定制、协议编辑、协议发送。协议调试器主流程图如图 6 所示，协议分析器流程图如图 7 所示。

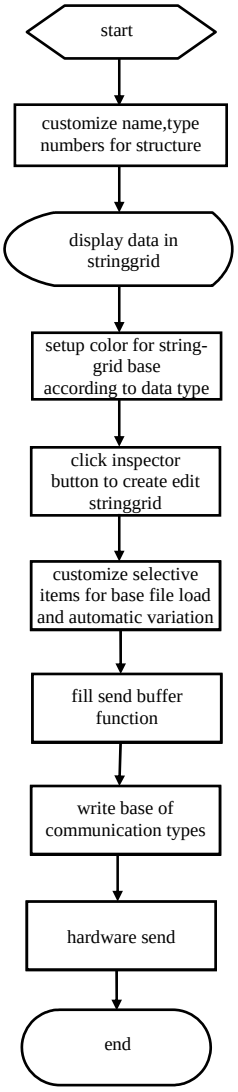

Fig.6 Protocol debugging flow
图 6 协议调试器流程图

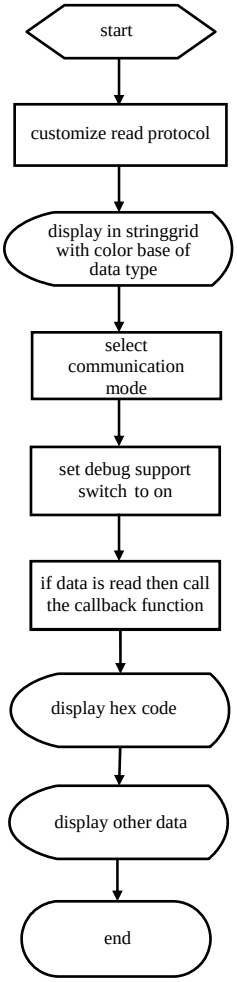

Fig.7 Protocol analysis flow
图 7 协议分析器流程图

不同数据类型使用不同颜色，有些数据类型需要十六进制输入(BYTE,WORD,DWORD)。该部分需要编写数据输入检查、范围检查等，为了给出十六进制方式下输入的明确标志，设计了格子的相应提示(三角小标志)[5]。

协议编辑设计是较复杂的，首先写一个缓冲区填写函数完成任意协议的缓冲区构造，该函数输入为界面的各元素，从界面上选择变量名称、变量类型、个数，构造出结构 B。函数输出为可编辑的内存数据，为了实现可编辑输出，使用了专业的字符串格子技术[6]， 因为通信协议是千变万化的(B 结构)，这部分编程较复杂，主要思路为：在可编辑字符串格子中取出数据，根据数据类型、数据个数，构造内存第一部分数据(协议结构中某个字段，例如速度，从第 3 字节开始，共 4 个字节)，进入循环再构造第 2 部分，直到循环完成。因为每一字段都是不同的，软件使用了诸多技术方法，这里略之。

当缓冲区填写函数调用后，调试器再调具体发送函数发出数据，发送函数由发送方式选择选出，使用什么方式发送数据将根据工程项目确定(硬件方式下使用串口或者网络，软件方式下使用进程间某通信方式)。在调试时，首先要确定接口方式(硬件初始化或者使用进程方式)。

协议的存储与装载功能：编辑过程的存储与装载也是常用的，当通信结构中元素(变量)较多时，这些功能就显得非常重要，分别使用内存流和组件的类函数实现该部分[7]。协议的定制如果可以存储，那么使用时就非常方便，一次定制可以多次使用。

对协议结构 B 中的整型数据和浮点数类型，调试器还设计了自动数据变化功能，协议中的某些数据可以来自某个数据文件的某列或者按照某个步长自动变化。该部分是该调试器最难设计的部分，例如需要模仿一个弹道的发送过程，设某协议规定：从通信结构的第 5 字节开始，每 4 个字节代表某个浮点数，共 3 个物理量 X,Y,Z，代表某个弹道数据，应用监控程序需要这组数据，例如弹道数据共 100 s，每秒由系统主机发送给某应用监控软件，每秒发送 20 次，当应用监控程序收到这些数据后，应转入数字引导工作方式，指挥某跟踪器(例如天线设备或者望远镜设备)按照该弹道去运动。使用该调试器代替系统主机，在协议编辑制作界面中将发送方式改为文件，首先按文件装载按钮将文件装入内存，在调试器界面上定制弹道数据分别位于文件哪一列，在协议结构中什么位置开始，然后启动定时发送功能，则调试器就自动发送一个变化的弹道数据给应用程序。设计了抽象的变化弹道

装入功能，变化数据可以多达 6 组。航天类应用程序经常需要检查某弹道装入后程序执行情况，这些内容往往需要另一通信方配合才能够检查，通过该调试器可以很方便模拟这个变化的弹道，它的自动定时发送会给应用程序送出这个变化的弹道，这样就提前解决了联试中的问题，相当于不需要对方配合就检查了你的监控程序编写的正确性，该部分因篇幅所限略之。

协议分析功能：按通信协议分析应用程序发送准确性，通常情况下接收端读到发送端的数据，经过解码后才知道是否准确。该调试器可以容易地实现该功能，在总调试界面上，设计了一个协议分析开关，一旦选中，调试器将接收到的数据按照通信读协议自动解析并显示，一方面可以看到发送的码字内存数据；另一方面可以看到具体的通信协议对应的字段数据。被调程序在未连接具体传感器之前，应用程序已通过调试器检查了发送数据的正确性[8]。

5　其他接口形式的调试方法

当系统中连入其他形式的板卡时，同样可以使用该调试器调试应用程序。假定有一块控制区域网络(Control Area Network，CAN)卡连入监控系统，在该卡未连入系统前，假定需要通信的协议数据结构已知，使用调试器调试 CAN 卡通信部分。注意该调试器并没有写 CAN 卡的通信函数，但仍可调试 CAN 卡。下面演示调试器使用文件方式代替 CAN 卡调试应用程序的方法。

设应用监控程序为进程 P1，调试器进程为 P2，应用程序同某硬件传感器接口为 CAN 卡，假定通信协议已知。现用调试器代替 CAN 卡，按通信协议 CAN 卡需要发送某个结构数据 B 每个数据给应用监控程序，调试器并没有 CAN 卡接口，但调试器进程可以产生出结构 B 内的各元素内存数据，并写为一个文件，再发出广播消息，告知所有进程该文件已经产生，如果应用监控程序能够读到这个广播消息，再在此时读到这个文件，就相当于从 CAN 卡读到了数据。

应用程序端需要稍作更改，以适应调试器对传来数据的响应，可以使用 2 种方式读该文件：手工方式，写一个按钮消息，读文件内容到 CAN 卡读入的内容缓冲区，代替 CAN 卡读缓冲区内容，相当于应用程序从 CAN 卡读到了数据；自动方式，在应用程序中重载 WinProG 过程，捕获由调试器的广播消息，在这个消息函数中，打开数据文件，读内容放在某个缓冲区，再将该内存数据复制到本该 CAN 卡读入的缓冲区，也相当于从 CAN 卡读到了数据。调试器在软件接口中设计了广播消息机制，当调试器一端的发送按钮按下后，调试器依据通信方式(软件方式)告知应用程序数据已经变化，并且给出变化的数据，这样应用程序如果设计了自动唤醒功能(捕获这个广播消息，接收数据)，那么应用程序就具备了与调试器通信的自动收发功能。

使用简单的文件方式调试 CAN 卡，假定应用程序读 CAN 卡形式化的语句为 ReadCan(设备号，读缓冲区指针，读长度)，读到数据后处理函数。

将 ReadCan 函数注释掉，改为读文件到这个缓冲区，则相当于从 CAN 卡读到数据；那么在真正的 CAN 卡未连到系统之前，应用程序无法调通这句代码。应用程序如果改为读文件，则可以调通，这样有关 CAN 卡的其他逻辑功能就可以全部试通，在未连接硬件 CAN 卡时已编写完成有关的 CAN 卡代码。同理，无论应用程序使用什么硬件接口，调试器都可以仿照这种方法调试。

调试器设计了多种软件接口的调试方式，为进程形式的方法，在单机调试时较有用，该方式下由调试器工具产生出开发者需要的内存数据，并通过进程通信方式传给应用程序，应用软件需要适当改造：加进程通信代码段；加广播消息机制以便具备自动醒知功能(调试器数据发生了变化，通过进程方式传给应用软件，同时发出了广播消息，被调试软件需要编写广播消息的过滤方法)。这就是自动方式。

协议的任意构造与分析功能是该设计的最大特点，因为它可以产生出程序员需要的任何内存数据，并且可以按照协议自动分析出另一个进程(应用监控程序)发出的数据准确性，这才是承研者最想得到的。协议的任意构造使用了多种技术方法，对各种输入都做了范围检查及异常处理，还做了对某些数据进行自动步长变化的特殊处理，这样调试器就可以自动发出某些变化数据。

6　结论

无论应用的工程项目属于哪个领域，只要有计算机控制部分(控制传感器或者计算机)，就存在协议通信问题。作者将这些问题进行了统一处理，设计了这款综合协议调试器，不再需要编写其他仿真代码即可帮助开发者实现工程项目软件的快速开发，在实验室环境下借助该调试器形成一个内环，提前调通应用项目软件，因为已经在实

验室环境下完整执行了项目软件，这样将在联调阶段节约大量时间。调试器工具在开发中可以帮助研制者快速调通整个项目，也可在联试时帮助分析员快速分离出通信中的问题，该调试器虽然是为航天类监控项目而设计，但是也可应用在各种与计算机通信有关的课题或者其他应用项目中。

参考文献：

[1] 赵晓玲. 可视化程序设计—Delphi[M]. 北京:机械工业出版社, 2005. (ZHAO Xiaoling. Visual programming-Delphi[M]. Beijing:China Mechanical Press, 2005.)

[2] 王小华. Delphi 程序员经验点滴桌面网络编程实例集锦[M]. 北京:兵器工业出版社, 2006. (WANG Xiaohua. Delphi programmers experience intravenous drip desktop network programming examples highlights[M]. Beijing:The Publishing House of Ordnance Industry, 2006.)

[3] 王艳平. Windows 网络与通信程序设计[M]. 2 版. 北京:人民邮电出版社, 2009. (WANG Yanping. Windows network and the communication program design[M]. 2nd ed. Beijing:Posts&Telecom press, 2009.)

[4] MARK E R,DAVID A S,ALEX Lonescu. 深入解析 Windows 操作系统[M]. 6 版. 北京:电子工业出版社, 2009. (MARK E R,DAVID A S,ALEX Lonescu. Deep parsing the Windows operating system[M]. 6th ed. Beijing:Publishing House of Electronics Industry, 2009.)

[5] 朱汉民. Delphi7 高级应用开发教程[M]. 北京:科学出版社, 2006. (ZHU Hanmin. Delphi7 advanced application development tutorial[M]. Beijing:Science Press, 2006.)

[6] 周爱民. Delphi 源代码分析[M]. 北京:电子工业出版社, 2004. (ZHOU Aimin. Delphi source code analysis[M]. Beijing: Publishing House of Electronics Industry, 2004.)

[7] 郑阿奇. Delphi 实用教程[M]. 北京:电子工业出版社, 2008. (ZHENG Archie. Delphi practical tutorial[M]. Beijing: Publishing House of Electronics Industry, 2008.)

[8] 苏春晓,王鹏,杨存榜,等. 基于网络的数字示波器远程集中软件平台[J]. 太赫兹科学与电子信息学报, 2014,12(6): 884-889. (SU Chunxiao,WANG Peng,YANG Cunbang,et al. Digital oscilloscope based on network remote centralized software platform[J]. Journal of Terahertz Science and Electronic Information Technology, 2014,12(6):884-889.)

作者简介：

蔡文斋(1962–)，男，安徽省宿州市人，硕士，高级工程师，主要研究方向为航天测控工程、工程控制类应用、软件测试.email:122790435@qq.com.