

ActiveX 在 LabVIEW 下的事件回调和数据传递

胡吉朝¹ 王定远² 安素方¹ 傅 玥¹

(1. 石家庄经济学院信息工程学院 石家庄 050031; 2. 西安交通大学机械学院 西安 710049)

摘 要:本文介绍了 ActiveX 控件的特点,阐述了 LabVIEW7.1 下 ActiveX 控件的使用原理及其事件回调的注册方法,解决了事件回调编程中的数据传递难题,保证了数据的实时传输和数据处理,避免了数据的丢失和数据覆盖。本文以 MSCOMM32 控件为例,介绍了 MSCOMM32 在 LabVIEW 下的事件回调和数据传递技巧,具有技术的通用性,对扩展 LabVIEW 自身功能具有应用价值和参考价值。

关键词:ActiveX; LabVIEW; 事件回调; 同步技术

中图分类号: TP3-05 **文献标识码:** B

Realization of ActiveX event callback and data transfer in LabVIEW

Hu Jichao¹ Wang Dingyuan² An Sufang¹ Fu Yue¹

(1. Department of Information Engineering, Shijiazhuang University of Economics, Shijiazhuang 050031;

2. Department of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049)

Abstract: The property of activeX control is introduced, the principle of using ActiveX control and the method of register event callback are both elaborated. And the problem of data transfer in event callback programming is solved, as a result, the real-time data transfer and data processing is ensured. The data lost or data covered is avoided. The MSComm32 ActiveX control is taken as an example to introduce how to program to solve the event callback and data transfer in LabVIEW7.1. The method is universal and common used, which can enlarge the LabVIEW itself.

Keywords: ActiveX; LabVIEW; event callback; synchronization technology

0 引 言

ActiveX 控件是一些软件组件或对象,常以 OLE 控件或 OCX 控件形式出现,可以将其插入到 WEB 网页或其它应用程序中,是微软作为 Internet 策略倡导的一种网络化多媒体对象技术。ActiveX 控件可以被大多数应用程序(组件容器)再使用,与具体的编程语言无关。目前,ActiveX 控件在 Windows 和 Macintosh 中运行。

当今多数应用程序开发软件都提供了对 ActiveX 技术的支持。在测控行业采用 ActiveX 技术是一种较新的尝试,它提供了跨语言编程的优良特性,大量的数据采集、数据处理、通信、视频等功能被封装成 ActiveX 控件提供给测控技术行业使用,减少了系统的开发难度,提高了代码的可继承性;IVI 技术标准也推出了 IVI-com 技术,实现仪器的可互换性操作和程序控制,对于计算机测控技术无疑将会起到巨大的促进作用。

NI LabVIEW 是虚拟仪器技术的缔造者和推动者,是

当今全球首推的虚拟仪器和面向计算机测控技术的开发环境,于 1986 年开始发行 1.0 版本,到目前为止经过了 20 年的更新,目前最新版本为 8.5,典型特点是数据流编程和 Express 技术。LabVIEW5.1 版本开始支持 ActiveX 控件的接口技术,LabVIEW7.0 版开始对 ActiveX 的接口技术进行了全新的改变,并提供了全新的 Express 技术。因此,如何采用 LabVIEW 新的 ActiveX 接口技术快速、高效的完成应用程序的功能实现和扩展,是一个难题和挑战,而在 LabVIEW 图形化编程环境中调用 ActiveX 控件所面临的一个最大难题是:如何解决 ActiveX 控件的事件回调并完成数据传递。

LabVIEW 处理 ActiveX 的接口方法和 VC++、VB 等并不相同,有些差别非常大,简单的模仿和翻译将会造成错误,因此,本文将详细探讨如何在 Windows XP 系统下采用 LabVIEW7.1 完成 ActiveX 控件的事件回调处理以及数据传递,以扩展 LabVIEW 自身的功能,并充分利用该技术加快计算机测控系统的开发,本文将所研究的内

作者简介: 胡吉朝(1979-),男,硕士研究生,助教,主要研究方向为网络和计算机应用技术。

容和方法将适合于 LabVIEW 7.0 及以后的版本。

1 ActiveX 在 LabVIEW 下的使用方法

NI 公司采用 ActiveX 技术开发了一系列的 LabVIEW 配套工具包,来扩展 LabVIEW 的专业化功能,比如:数据库连接工具包调用了微软的 ADO ActiveX 控件,报表生成工具包调用了微软的 Word 和 Excel 的 ActiveX 控件。这些工具包的底层技术就是 ActiveX 控件技术^[1-2]。

1.1 ActiveX 控件的组成

ActiveX 主要由自动化服务器、自动化控制器、控件、文档组成。

自动化服务器是可以由其他应用程序编程驱动的组件。自动化服务器至少包括一个,自动化服务器可以含有也可以没有用户界面(UI),这取决于服务器的特性和功能。

自动化控制器是那些使用和操纵自动化服务器的应用程序。

控件等价于以前的 OLE 控件或 OCX。一个典型的控件包括设计时和运行时的用户界面、属性、方法和事件,但有的控件可能只存在一个部分。属性是对具体对象参数或特征的定义,方法是启动一种函数的请求,事件是来自一个对象的函数调用。

文档表示一种不仅仅是简单控件或自动化服务器的对象。ActiveX 文档在结构上是对 OLE 链接和模型的扩展,并对其所在的容器具有更多控制权。一个最显著的变化是菜单的显示方式。一个典型的 OLE 文档的菜单会与容器菜单合并成一个新的集合,而 ActiveX 文档将替换整个菜单系统,只表现出文档的特性而不是文档与容器共同的特性。

容器是一个可以作为自动化服务器、控件和文档宿主的应用程序。

1.2 LabVIEW 下调用 ActiveX 控件

ActiveX 采用 C/S 方式与编程环境通信,ActiveX 通常作为服务器端。操作 ActiveX 控件,最主要的就是要使用好它的属性、方法函数,并做好中断事件的回调处理。LabVIEW 可以作为 ActiveX 自动控制服务端(Server)和客户端(Client)。当 LabVIEW 作为 ActiveX 自动控制客户端时,它可以控制其他的 ActiveX 服务端的应用程序,这时 LabVIEW 能够从这些应用程序中得到它们的属性和执行方法并对它们进行设置^[1]。

LabVIEW 7.1 专门针对 ActiveX 设置的控件包含了 8 个功能函数:Automation Open、Close Reference、To Variant、Variant to Data、Invoke Node、Property Node、Register Event Callback、UnRegister Event Callback。如图 1 所示。

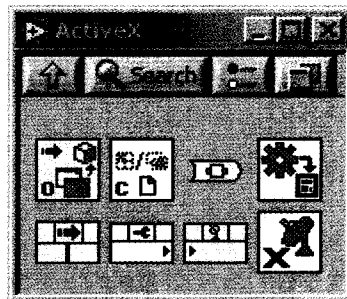


图 1 ActiveX VIS 控件

在 LabVIEW 中创建一个调用 ActiveX 控件的客户端应用程序,可以分为以下几个步骤:

(1)使用 Automation Open 函数打开一个自动化控制标识符(如 Excel 程序对象);

(2)用 Property Node 设置或得到对象的属性;

(3)用 Invoke Node 执行属于这个对象的方法;

(4)如果需要,还应使用数据转换函数 Variant to Data 把 ActiveX 格式的数据转换成 LabVIEW 支持的格式的数据,使用 To Variant 将 LabVIEW 支持的数据类型转换成 ActiveX 控件使用的变体类型;

(5)如果该 ActiveX 控件具有事件,则通过 Register Event Callback.vi 注册事件,并完成事件回调的处理,采用 UnRegister Event Callback.vi 注销事件注册,释放资源;

(6)用 Automation Close 函数关闭自动控制标识符。

1.3 ActiveX 事件回调的注册和注销

在 LabVIEW 中操作 ActiveX 的事件中断函数是通过 Register Event Callback.vi 节点来实现的,使用这个节点需要先对它进行配置。编程时,先把它的 Event 管脚与 ActiveX 控件的 Refnum 端相联接,这时 Register Event Callback 节点就会含有 ActiveX 的所有中断函数信息。

为了实现事件回调,就得设定并配置事件的中断服务程序。关键之处就是需要提前定义数据传递方式,也就是在配置中断服务之前需要提前定义好采用何种方法进行数据传递,不同类型的方法将导致不同的中断回调 VI。

提前把队列的 Reference 输出和 User Parameter 连接,然后右键点击 Register Event Callback 节点的 VI Ref 管脚,选择 Create Callback VI,则程序自动生成一个对应的中断回调的子 VI 和 VI Ref 管脚相联。然后,双击这个子 VI,进入其程序框图,编辑对应的中断服务程序,完成中断所对应的功能,并保存该 VI。当程序结束时应该采用 UnRegister For Events 注销对应的中断注册,以释放资源。

1.4 ActiveX 事件的数据传递

注册的事件有的无需返回数据到主程序,此类事件往往在对应的事件回调中通过调用方法函数或者设置属性即可;而一些事件回调需要向主程序中返回数据,以提供

给主程序进行显示或者处理。为了能够实时的把事件回调 VI 中的数据传递到主程序必须考虑好 ActiveX 的事件回调的数据传递问题。

中断服务和主程序依靠中间的数据读写产生联系,这个联系是采用了数据的非连线传递方法^[3]。在 LabVIEW 8.0 版本之前可以通过:全局变量、VI Server、队列、通告、信号灯、事件、邮箱、DSTP、网络协议等方式来实现非连线的数据传递;LabVIEW 8.0 版本之后增加了共享变量技术可以代替全局变量完成本机或网络的数据传递。

通常的面向对象的编程软件如 VC++、VB 等在处理单线程程序的回调事件数据到主程序的数据传递时采用全局变量的方法^[4]。LabVIEW 可以借鉴全局变量的思想完成数据传递,但要保证好全局变量的“一写多读”原则,避免数据的读写冲突。文献[4]采用全局变量的思想完成了 MSComm32 的回调事件数据传递。

LabVIEW 天然具备了多线程、多任务的执行机构特点,编程简单,快捷。测控系统中任务相对比较复杂,为了实现高效的程序管理,往往要依靠多线程、多任务,通常最简单的办法是多 while loop 的并行执行结构。在非连线的 while loop 间无冲突的传递数据并能完好的实现多任务、多线程、并行、同步处理的方法最常用的是控件参考属性(VI control Reference)、通告(Notification)、队列(Queue)、事件(Occurrence)、信号灯(Semaphore)、集合(Rendezvous)同步技术。把这些技术和方法用在 ActiveX 的事件回调数据传递上,完全可以实现所需的功能。在这里,推荐使用控件参考属性、通告或者队列的方式来实现回调事件中的数据到主程序的数据传递和同步更新。

队列(Queue)可作为 while loop 间或 VI 间的数据管道,它相当于一个 FIFO,利用它来弹性调节 while loop 或 VI 之间的速度差,避免了采用全局变量而可能引起的读写冲突而造成的数据覆盖或数据丢失的现象^[1]。以队列实现 MSComm32 控件事件回调的数据传递为例说明。详细使用方法可参考文献[5]。

在 Tools—>Advanced—>Import ActiveX Controls 下将 Microsoft 的 MSComm32 控件“Microsoft Communications Control, Version 6.0”添加到 LabVIEW 环境中,存放在缺省路径。将 WinSock 控件放置在 Front Panel 上,可以查看其属性^[5]。如图 2 所示。

LabVIEW 切换到框图程序环境,找到 MSComm32 控件,在 Block Diagram 中按要求设置控件的属性,完成控件的初始化,并完成事件回调的注册和中断服务程序的编写,主程序进行统一的管理和数据处理,实现所需串口通信功能。实际操作时,一个串口只能有一个 MSComm32 控件与之相对应。下面将着重探讨事件回调的注册和数据传递。

程序中先创建一个队列,并把这个队列的 Reference 输出和 Register Event Callback 所注册的事件的 user parameter 相连接,即可看到 MSComm32 控件只有一个 On-

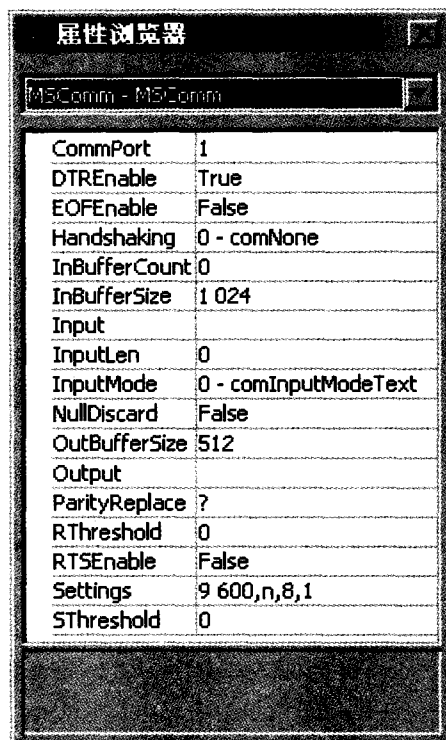


图 2 MSComm32 控件属性

Comm 事件,右击该事件对应的 VI Ref 处,选择 Create Callback vi,程序即可自动生成一个事件回调的程序,可以称作中断服务程序。如图 3 和图 4 所示。

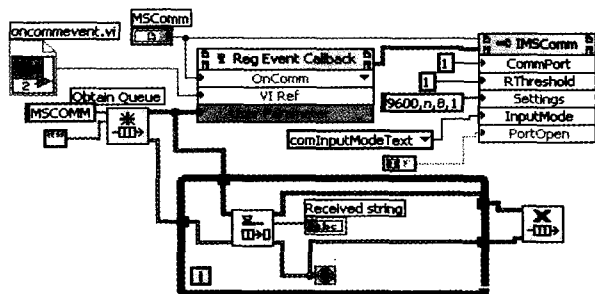


图 3 MSComm32 控件的事件数据传递

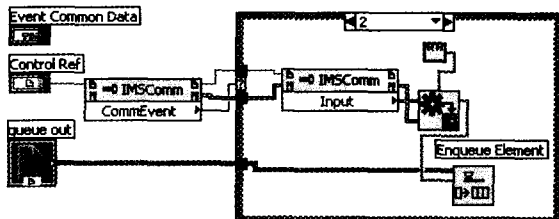


图 4 MSComm32 的 OnComm Event callback

在中断服务程序中引出回调事件的控件 VI Ref,并利用 MSComm32 控件的属性函数 input 进行中断服务的数据提取和接收,并接收到的变体类型数据采用 Variant to Data 转变成 LabVIEW 支持的 string 类型,

。(下转第 23 页)

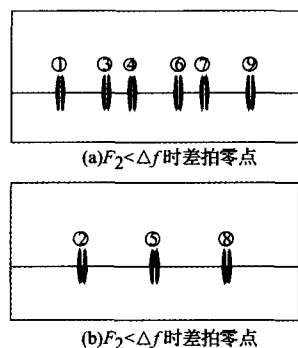


图6 示波器波形图

测量方法:

(1)调整低频振荡器的频率,示波器屏幕上的差拍零点会两两相互靠近或两两相互离开;

(2)增大低频振荡器的频率,图6(a)中的差拍零点①-③、④-⑥、⑦-⑨相互靠近,直至最后合并为图6(b)中的②、⑤、⑧。此时应仔细调整低频振荡器的频率,以得到差拍零点的准确值;

(3)读取电子计数器的读数 F = 调频频偏值 Δf ;

(4)若调频频偏不对称,继续再增大低频振荡器的频率时,还会出现新的差拍零点。

测量误差^[4]:

(设:调频波载频 $f_1 = 30$ MHz,调制频率 $F = 150$ Hz,调频频偏峰值 $\Delta f = 20$ kHz)

$$\frac{\Delta F}{\Delta f} = \pm \frac{\text{被测频率 } F \times \text{频率计时基误差} \pm 1 \text{ 计数值}}{\Delta f} \times 100\% = \pm \frac{150 \times 10^{-7} \pm 1}{20 \times 10^3} < 60.1\%.$$

(注: E-312 频率计的时基误差 $= 61 \times 10^{-7}$)。

特点:

- (1)测量误差小,小于 61%,
- (2)可精确的测量出调频频偏的不对称性。

4 结束语

著名的贝塞尔函数调零法是精确测量调频波调频频偏的良好方法,但是,它不能精确测量调频指数 $m \leq 1$ 及 $m \geq 100$ 调频波的调频频偏。因为贝塞尔调零法是根据 $J_0(m) = 0$ 的根值推算调频信号的调频频偏值,而贝塞尔函数 $J_0(m)$ 的第一个根值 m 为 2.404 8 (最小值),其它根值 m 均大于 2.404 8。所以,不能精确测出调频指数 $m \leq 1$ 调频波的调频频偏值。另外,由于贝塞尔函数的峰值随着变量 m 的增加而减小,当调频指数增大到 $m \geq 100$ 时, $J_0(m)$ 的峰值非常小,很难确定其根值(零点)位置,所以亦无法精确测出调频指数 $m \geq 100$ 调频波的调频频偏值。本文提出的“载频移相比较法”及“频偏峰值零拍法”弥补了“贝塞尔函数调零法”的不足,可以精确测量调制指数 $m \leq 1$ 及 $m \geq 100$ 调频波的调频频偏。文章对“载频移相比较法”及“频偏峰值零拍法”的工作原理及测量误差进行了详细的理论分析,说明了该两种方法可以精确测量调制指数 $m \leq 1$ 及 $m \geq 100$ 调频波的调频频偏的可行性,并且做出了产品,赋予实践。

参考文献

- [1]郑家祥. 电子测量基础[M]. 北京:国防工业出版社,1981.
- [2]张肃文. 高频电子线路[M]. 北京:高等教育出版社,1991.
- [3]潘志. 数学手册[M]. 北京:中国矿业大学出版社,1995.
- [4]王谨之. 常用电子测量技术手册[M]. 天津:天津科学技术出版社,1989.

(上接第9页)

并通过队列发送。主程序的接收队列数据的函数放在 1 个单独的 while loop 内,当接收到队列数据提取出数据,否则该 while loop 处于休眠状态,这样就保证了不消耗内存,且能最大限度的调整循环间的数据收发同步,不丢失或覆盖数据。

2 结束语

采用 LabVIEW 调用 ActiveX 控件可以充分利用已有的代码函数并扩展 LabVIEW 软件的自身功能,从而美化应用软件的外观和拓展在测控技术领域的应用;针对事件回调和数据传递,可以将 LabVIEW 的数据流技术和事件中断技术完美结合起来,更好的利用计算机的 CPU。随着计算机技术和测控技术的多方位结合,面向计算机的计算机控制及其自动化技术将能依托 LabVIEW 的 G 语

言优势,更好的得到发展。

参考文献

- [1]钟凡亮,严国萍. LabVIEW 平台下的测试仪软件系统设计与实现[J]. 计算机与数字工程,2007,35(1):138-140.
- [2]汪翠英,裴锋. LabVIEW 中 Excel 报表生成功能开发[J]. 仪表技术,2005(3):60-64.
- [3]杨乐平,李海涛. LabVIEW 高级程序设计[M]. 北京:清华大学出版社,2003.
- [4]李现勇. Visual C++ 串口通信技术与工程实践[M]. 2 版. 北京:北京人民邮电大学出版社,2004.
- [5]王定远,胡吉朝,李媛. 基于 MSComm32 和 LabVIEW 的串口通信技术[J]. 国外电子测量技术,2006,25(4):61-64.