空 军 工 程 大 学 学 报(自然科学版)
第1卷第1期 JOURNAL OF AIR FORCE ENGINEERING UNIVERSITY Vol.1 No.1
2000年4月 (NATURAL SCIENCE EDITION) Apr. 2000

基于VISA库的VXI总线编程

肖明清， 周越文
(空军工程大学 工程学院航空兵器工程系，陕西 西安 710038)

摘 要： 介绍了利用VISA库进行VXI总线编程的几个要点，包括仪器的寻址、消息级器件的访问、寄存器级器件的访问、异步事件处理和出错处理等，并给出了示例程序。

关键词： VISA库；SCPI命令；VXI总线
分类号： TP274^{+}5 **文献标识码：**A **文章编号：**1009-3516(2000)01-0055-04

进行VXI总线编程离不开VISA库的支持。VISA库以动态链接库(DDL)的形式向Visual Basic，Visual C++等程序设计语言提供了访问VXI总线资源的手段。VISA是虚拟仪器软件结构(Vitual Instrument Software Architecture)的缩写，实质是一个输入输出接口软件库及其规范的总称。它是由VXI总线即插即用联盟提出和规定使用的，其目的是使不同厂家的仪器具有相同的编程接口。

控制消息级器件要用到SCPI命令，VISA库的重要功能之一就是向仪器发送SCPI命令。SCPI是可编程仪器标准命令(Standard Commands for Programmable Instraments)的缩写，是一种用来控制仪器的命令语言。它用一种标准的方式来描述各种各样的仪器功能，规定了在控制器到仪器和仪器到控制器之间信息交换层消息的构造和内容，使各类不同的仪器更容易被理解和使用，减少了自动测试系统开发的工作量和时间。

1 仪器的寻址

在与一个仪器通讯之前，必须将测试应用程序与仪器进行连接，即要建立仪器与应用程序之间的通讯途径，这种通讯途径被称之为Session，确切地说，Session就是应用程序与仪器之间通讯途径的一个实例(instance)。

在程序中建立一个与某仪器进行通讯的Session需调用VISA库的两个函数viOpenDefaultRM和viOpen。其使用方法见程序例1。

2 访问消息级器件

消息级器件具有自己的微处理器，能够解释字符串形式的SCPI命令。SCPI命令具有较强的可读性，因此编出的测试程序容易理解。VISA输入/输出库函数提供了向仪器发送SCPI命令的能力。viRead和viWrite是两个最基本的VISA输入/输出库函数。viWrite的形式为：

viWrite(vi,buf,count,retCount)

参数vi代表某仪器的一个句柄；buf是要发送的字符串(一般是SCPI命令)；count是要发送的字符总数；retCount返回实际发送了的字符数。该函数把字符串buf发送到vi所代表的仪器。viRead的形式为：

viRead(vi,buf,count,retCount)

参数vi是仪器句柄；字符串变量buf用来存储从仪器读回的数据；count是欲读取的字符总数；retCount返

收稿日期：1999-12-17
作者简介：肖明清(1963-)，男，教授.

回实际读取的字符数。程序例 1 演示了这两个函数的用法，该程序使某万用表进行交流电压测量，并显示测量结果。

[程序例 1]

```
Sub Main()
Dim sesn As Long,vi As Long,Buffer As String*32,retCount As Long,Cmd As String,err As Long
'为了与 VXI 总线上逻辑地址为 32 的数字多用表通讯，先建立相应的 session
err=viOpenDefaultRM(sesn)
err=viOpen(sesn,"VXI::32::INSTR",VI_NULL,VI_NULL,vi)
'使议器复位
Cmd="*RST"+Chr$(10)
err=viWrite(vi,Cmd,Len(Cmd),retCount)
'使仪器测量交流电压
Cmd="MEAS:VOLT:AC?"+Chr$(10)
err=viWrite(vi,Cmd,Len(Cmd),retCount)
'读取测量结果
err=viRead(vi,Buffer,32,retCount)
'显示测量结果
MsgBox Buffer,0
'关闭仪器 session
err=viClose(vi)
err=viClose(sesn)
End
End Sub
```

3 访问寄存器级器件

访问消息级器件的方法除了上节所讲之外，还可以直接读写器件的寄存器，但后者不太常用。而对于寄存器级器件的访问就只能是直接读写其寄存器。VISA 提供了两种直接对寄存器进行编程的方法：高级存储器函数和低级存储器函数。

高级存储器函数包括 viIn、viOut、viMove 和 viMoveOut，前两个用于传送单个数据，后两个用于传送数据块。由于篇幅所限，本文只介绍前两个的用法。ViIn 对寄存器进行读操作，其形式为：

viIn(vi,space,offset,val)

viOut 对寄存器进行写操作，其形式为：

viOut(vi,space,offset,val)

程序例 2 演示了 viIn 与 viOut 的用法，该程序先读取了位于 VXI::144 处的某寄存器级器件的识别(ID)寄存器和器件类型寄存器，接着向相对地址为 24 的寄存器写入了 0AAAAH。

[程序例 2]

```
Sub main()
Dim Sesn As Long,vi As Long,err As Long,id_reg,devtype_reg As Integer,out_data as Integer
Dim msg As String
err=viOpenDefaultRM(sesn)
err=viOpen(sesn,"VXI::144::INSTR",VI_NULL,VI_NULL,vi)
'读仪器的识别寄存器
err=viIn16(vi,VI_A16_SPACE,0,id_reg)
'读仪器的器件类型寄存器
err=viIn16(vi,VI_A16_SPACE,2,devtype_reg)
```

```
    '以十六进制数显示两寄存器的内容
    msg="ID register=&H"+Hex(id_reg)+Chr$(10)
    msg=msg+"Device Type Register=&H"+Hex(devtype_reg)
MsgBox msg,0
    '将十六进制数 0AAAAH 写入仪器中相对地址为 24 的寄存器
    out_data=Val(&HAAAA)
    err=viOut16(vi,VI_A16_SPACE,24,out_data)
    err=viClose(vi)
    err=viColse(sesn)
    End
    End Sub
```

低级存储器函数包括 viMapAddress，viUnmapAddress，viPeek 和 viPoke，用低级存储器函数读写寄存器首先要用 viMapAddress 进行存储器映像，然后用 viPeek，viPoke 进行读写操作，最后要用 viUnmapAddress 清除映像。可见低级存储器函数在使用上比高级存储器复杂，但其优点是执行速度要快得多。

4　异步事件处理

异步事件包括服务请求(SRQ)、中断和硬件触发。由于篇幅所限，这里仅对硬件触发事件的处理举一个例子：程序例 3。该程序中所用到的有关函数请参阅文献[1]或其它有关 VISA 库的手册。

[程序例 3]

```
sub main()
Dim sesn As Long,vi As long,err As Long,eventType,eventVi As Long,msg As String
Dim trigId As Integer
err=viOpenDefaultRM(sesn)
err=viOpen(sesn,"VXI::0::INSTR",VI_NULL,VI_NULL,vi)
'选择触发线 TTL0
err=viSetAttribute(vi,VI_ATTR_TRIG_ID,VI_TRIG_TTLO)
'使能触发事件
err=viEnableEvent(vi,VI_EVENT_TRIG,VI_QUEUE,VI_NULL)
'执行触发
err=viAssertTrigger(vi,VI_TRIG_PROT_SYNC)
'等待事件出现
err=viWait0nEvent(vi,VI_EVENT_TRIG,10000,eventType,eventVi)
If err=VI_ERROR_TMO Then
'如在设定时间内未接收到事件则程序执行至此
msg="Timeout 0ccurred! Event not received."
Msgbox msg,0
Else
'检查触发信号出现在哪条触发线上
err=viGetAttribute(eventVi,VI_ATTR_RECV_TRIG_ID,trigId)
msg="Trigger Event 0ccered!"+Chr$(10)
Select Case trigId
    Case VI_TRIG_TTL0
      msg=msg+"Trigger that fired:TTL0"
    Case Else
      msg=msg+"Trigger that fired:<other>"
End Select
```

```
    MsgBox msg,0
    viClose(eventVi)
End If
err=viClose(vi)
err=viClose(sesn)
End
End Sub
```

5 出错处理

前面所给的几个程序只是为了演示 VISA 库的有关函数的功能，没有包括出错处理，然而任何一个有实用价值的测试程序都应该具有出错处理部分，这是一种良好的编程习惯。

在测试应用程序的执行过程中，可能出现的错误有两种，一种是 VISA 库函数错误，一种是仪器错误。

欲捕获 VISA 库函数错误应在每次 VISA 库函数调用之后检查其返回的函数值。假设返回值为 err，则当 err=VI _ SUCCESS 时表明函数执行成功，无错误出现；当 err<VI _ SUCCESS 时表明出现了致命性错误；当 err>VI _ SUCCESS 时表明出现了警告性错误。

在每一次仪器操作之后应检查仪器以确保无错误出现，检查仪器错误的方法是向其发送 SCPI 命令“SYSTEM:ERR?”。

参 考 文 献

[1] VPP-4.3-1995 The VISA Library[S].

[2] IEEEstd 1155-1992 IEEE Standard For VMEbus Extensions For Instrumentation:VXIbus[S].

VXIbus Programming Based on VISA

XIAO Ming-qing, ZHOU Yue-wen

(Dept. of Aeronautical Weapon Engineering of the Engineering Institute, AFEU., Xi'an 710038, China)

Abstract: The main points about development of application programs based on VISA for VXIbus are presented, including addressing an instrument, accessing a message-based device and a register-based device, handling events and errors. Some practical programs are presented.

Key words: VISA; SCPI; VXIbus