

文章编号:1001-9227(2001)05-0062-03

## LabVIEW 与 C 语言的混合编程

李俊 陈湘波\*

(中国科学院等离子体物理研究所 合肥,230031)

(\*合肥工业大学电气工程学院 合肥,230009)

**摘要:** LabVIEW 是一种优秀的虚拟仪器系统开发平台,广泛应用于工控开发,而 C 语言是目前广泛使用的功能强大的编程语言。本文介绍了 LabVIEW 与 C 语言的混合编程方法,并给出了编程实例。

**关键词:** LabVIEW C 语言 混合编程

**ABSTRACT:** LabVIEW is a excellent dummy instrument system,abroadly applied to project empolder. And C language is strong and used in all aspects. This paper introduced their mixing programming technique and presented examples.

**KEYWORDS:** LabVIEW C language Mixing programming

**中图分类号:** TP31

**文献标识码:** B

### 0 引言

LabVIEW(Laboratory Virtual Instrument Engineering Workbench)是美国国家仪器公司(简称 NI 公司)推出的可视化的虚拟仪器系统开发平台。它是一种基于 G 语言(graphical programming language)的革命性的图形化开发平台,主要用于数据的采集、分析、处理和表达。

LabVIEW 中包含了丰富的子程序和函数库,如数据采集、信号处理、概率统计及 GPIB、VXI 等各种程序库,利用这些函数和子程序,用户可快速建立起自己的采集控制系统,但是在很多系统中,从采集的数据到输出的控制信号之间需要经过大量和复杂的数学计算,而这时如果再用 LabVIEW 本身的图形语言——G 语言来编写计算程序,往往过多的连线方式会使人眼花缭乱,而且不利于修改和改进,而用传统程序语言的文字编程方式则显得更简洁、高效。与此同时,在某些场合下,使用别种语言可能会获得更高的程序执行效率,如在某些数据采集中,对时间要求比较严格或有大量的数据操作等,这时用 C 语

言可能会更好。CIN(Code Interface Node)和 DLL(Dynamic Link library)是 LabVIEW 调用外部语言代码的两种方法。下面我们一一介绍。

### 1 CIN 节点的使用

#### (1) CIN 节点的建立

在块图表表单中 CIN 表现为一个具有输入输出端口的图标。



用可将需要调用的 C 代码编译成 LabVIEW 所能识别的代码格式后和此节点相连。当程序执行到此节点时,LabVIEW 将会自动调用与此节点相关联的外部代码,并向 CIN 传递特定的数据结构。用户可以向 CIN 传递任意复合的数据结构。LabVIEW 中数据的存储格式遵循了 C 语言的数据存储格式,二者完全相同。LabVIEW 5.1 支持的编译器有:Visual C/C++、Microsoft C/C++、Win32 Microsoft SDK (Software Developer's Kit) C/C++、Watcom C/386 for Windows 3.1 等。

下面举一个 CIN 在计算中的用法来说明:

假如我们要计算  $a + b = c$  ( $a, b$  为输入,  $c$  为输出), 打开 LabVIEW, 在 Panel 上选二个 Digital control 和一个 Digital indicator, 如图 1。再在 diagram 窗口中, 从 Functions 面板上选取 CIN 到窗口中, 根据参数和结果的数目决定 CIN 上 terminal 的多少, 并将连线连好, 如图 1, 然后右键点击 CIN, 从弹出的菜单中选中 Create c File 这一项, LabVIEW 就会自动为你在特定的目录下生成 C 源文件, 如下所示:

```
/ *
    * CIN source file
*/
#include "extcode.h"

CIN MgErr CINRun (float64 *Numeric ,float64 *
Numeric . 2 ,float64 *Numeric . 3) ;

CIN MgErr CINRun (float64 *Numeric ,float64 *
Numeric . 2 ,float64 *Numeric . 3)
{
    / * ENTER YOUR CODE HERE */
    return noErr ,
}
```

其中 extcode.h 定义了基本的数据类型和一些 CIN 要使用的例程 (routines), CIN 有八个基本例程: CINRun, CINLoad, CINSave, CINUnload, CINAbort, CINInit, CINDispose, CINProperties; 其中 CINRun 是必需的, 它当程序执行到 CIN 时被调用执行, 接收输入值和输出结果; 其他的例程可用来执行一些特别的任务, 如想在 VI 第一次被装入时执行某些操作就可调用 CINLoad 例程, 将操作代码放入例程内即可。

现在我们来看上例, 将计算的代码输入 CINRun 函数体内, 在头部加上 Math.h, 整个代码如下:

```
/ *
    * CIN source file
*/
#include "extcode.h"
#include "math.h" c 语言里用于计算的头文件

CIN MgErr CINRun (float64 *Numeric ,float64 *
Numeric . 2 ,float64 *Numeric . 3) ;
```

```
CIN MgErr CINRun (float64 *Numeric ,float64 *
Numeric . 2 ,float64 *Numeric . 3) ;
{
    / * ENTER YOUR CODE HERE */
    *Numeric . 3 = *Numeric + *Numeric . 2 ;
    return noErr.
}
```

当然, 如果你不喜欢它的参数, 你也可将它们换成别的标识, 如把 \*Numeric 换为 a 等等, 下面说说 CIN 的编译。

## (2) CIN 节点原代码的编译

程序编好了以后, 我们就需要对它进行编译, 上面提到, 有多种编译器, 也可在多种平台上编译, 下面我们以 VC 6.0 来说明:

a) 打开 VC 6.0, 选中 File - > New..., 在项目类型中选 Win32 Dynamic - Link Library, 创建一个新的 Dll 项目, 存在自己的目录中, 如 d:\ test;

b) 把 LabVIEW 下的 Cintools 目录拷到任一个根目录下, 这是因为在编译时, 包含 Cintools 目录的路径不能有空格, 而通常 Cintools 目录是存在 \ Program Files \ National Instruments \ LabVIEW 下的。这里, 我们把它拷到 d:\ cintools. 选中 Project - > Add To Project - > Files..., 把 Cintools \ Win32 目录里的 cin.obj, labview.lib, lvsb.lib, lvsbmain.def 和刚编写的 C 源程序一起加到项目中, 这些都是创建 CIN 所必不可少的;

c) 把 Cintools 加进路径中。选中 Project - > Settings..., 把 Settings for 设为 All configurations, 再选中 C/ C++ 标签, 将 Category 设为 Preprocessor, 在 Additional include directories 字段填上包含 Cintools 目录的路径。本例中路径为 d:\ cintools;

d) 选中 Project - > Settings..., 把 Settings for 设为 All configurations, 再选中 C/ C++ 标签, 将 Category 设为 Code Generation, 将 Struct member alignment 字段设为 1 byte, Use run - time library 字段设为 Multi-threaded Dll;

e) 定制输出: 选中 Project - > Settings..., 把 Settings for 设为 All configurations, 再选中 Custom Build 标签, 在 Commands 和 Outputs 字段分别填上 < your path to cintools > \ win32 \ lvsbutil S(TargetName) - dS (WkspDir) \ S(outDir) S(TargetName) . lsb 这两个宏, 本例中的 < your path to cintools > 代替为

d:\cintools。

现在开始编译动态连接库,就会在你的项目 Debug 文件夹中产生一个后缀名为 .lsb 的文件,这正是我们所需要的。打开 LabVIEW,在 diagram 里,右键单击 CIN,在弹出的菜单中,选中 Load Code Resource,将 .lsb 文件装入即可。当然,以上例所举的简单相加计算是不能显示 CIN 的优点的,但如果你有一个上千行的复杂计算的话,你就会发现 CIN 的简洁与灵活性。

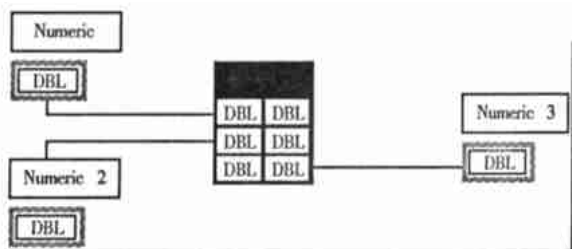


图 1

## 2 DLL 的使用

动态链接库 (DLL) 是一个包含许多函数的可执行模块,在 LabVIEW 里使用 DLL 和在其他语言里使用 DLL 一样简单,下面举个例子来说明 DLL 在 LabVIEW 里的用法。我们知道,NI 公司生产的采集卡是很昂贵的,但在平常的生产应用中,我们有时只需低廉的普通采集卡就会有很好的效果,而通过使用 DLL 的形式,我们就能够在 LabVIEW 里驱动一般的采集卡。下面以一段简单的控制继电器开合的程序为例,还以 VC 6.0 作为开发平台:

打开 VC,新建一个 MFC AppWizard (dll),假如名为 test。

在工程里,打开 test.cpp 文件,在文件尾部加上自己的函数,假如名为 controljdq,编写函数如下:

```
extern "C" __declspec(dllexport) controljdq(int time)
{ int time;
```

```
    int baseA = 640; 设置采集卡板基地址
```

```
    _outp(baseA + 5, 0x01); 向数字量信号输出寄存器写入 1,寄存器最低位输出高电平,经放大电路控制继电器闭合
```

```
    Sleep(time * 1000); 控制继电器闭合的时间
```

```
    _outp(baseA + 5, 0); 寄存器最低位输出低电平,继电器断路
```

}

接下来,选中 Build test.dll 即可(别忘了加上 #include "conio.h" 在文件头),这时在你的 test 目录下就会生有 test.dll 文件。下面,让我们看一下在 LabVIEW 里如何使用这 dll 文件。打开 LabVIEW,选中 Call Library Function 图标到 Diagram 窗口中,右键点击图标,在弹出的菜单中,选中 Configure... 命令,在随后弹出的对话框中,输入刚才编译好的 dll 文件名和要使用的函数名,这里为 controljdq,选好要的参数个数,参数类型,参数的传递顺序及函数的返回类型,点 OK 按钮即可。如图 2 所示。



图 2

## 3 结束语

通过以上实例,我们体会到 CIN 和 DLL 在 LabVIEW 中的作用,它把 LabVIEW 与 C 语言的优点集合到了一起,把 C 语言的高效和 LabVIEW 的丰富界面融合了起来,弥补了 LabVIEW 的不足之处,丰富了 LabVIEW 的应用。C 语言是目前广泛使用的功能强大的编程语言,通过与 C 语言的接口 LabVIEW 大大扩展了自身的功能。我们实验组在解决天线阻抗测量中,便是利用了 CIN 在计算中的功能,将计算程序代入 CIN 中,解决了天线阻抗计算问题,并使阻抗信号采集和计算程序融为一体。

## 参考文献

- 1 LabVIEW User Manual for Windows. National Instruments Corp., 1998