

文章编号: 1005-0523(2004)02-0080-05

# Lab Windows/CVI 在测控系统中的应用及其接口实现

刘立月<sup>1</sup>, 汤文亮<sup>1</sup>, 傅军栋<sup>2</sup>

(1. 华东交通大学 信息学院, 江西 南昌 330013; 2. 华东交通大学电气与电子工程学院, 江西 南昌 330013)

**摘要:**以 PCL-818 卡软件开发为例, 阐述 Lab Windows/CVI 开发试验与测试系统测试控制软件全过程; 针对 VC 与 Lab Windows/CVI 的消息机制冲突问题, 提出了类似进程封装思想, 并实现了 VC++ 开发环境下对 Lab Windows/CVI 开发模块的有效调用。

**关键词:** Lab Windows/CVI; VC++ 开发环境; 试验与测试系统

**中图分类号:** FP334

**文献标识码:** A

## 1 概述

试验与测试应用系统的具有自身特点: 底层的硬件通信能力、仪器控制能力、软件运行对时间的严格要求和专业化的信息分析处理等。NI 公司 Lab Windows/CVI 开发平台主要是针对试验与测试应用系统的自身特点而推出的, 相对其它开发工具 (VC、VB、PB 等) 而言, Lab Windows/CVI 优点在于硬件的控制及数据的处理能力上, 而在数据库、网络应用开发能力不及其它开发工具。因此, 可以利用 Lab Windows/CVI 开发试验与测试系统测试控制软件, 而用 VC 开发数据库与网络应用, 以有效实现试验与测试系统数据库管理及网络通讯。

## 2 Lab Windows/CVI 在测控系统中的应用

Lab Windows/CVI 是 NI 公司开发的基于标准 C 语言的软件可视化、交互式的开发工具, 具有标准 Windows 风格的操作界面, 内含功能齐全的软件工具包 (仪器控制、I/O 控制、通讯、数据处理) 等, 有很强的数据处理、数据分析功能; 可运行与 Win98/WinNT/Unix/Win2000/WinXP 等系统中, 适用于自动

测试、自动控制、测试仪器通信、测试硬件控制和信号分析处理的软件开发中。随着其功能不断提高, Lab Windows/CVI 已经成为测试控制界最受欢迎的软件开发工具。

### 2.1 Lab Windows/CVI 的特点

Lab Windows/CVI 作为测试控制和信息处理领域的一个优秀的开发平台, 从软件开发的角度来看, 具有以下一些特点:

- 1) 基于标准 C 语言, 易于学习与掌握;
- 2) 可视化、交互式的开发工具, 标准 Windows 风格的操作界面, 包含大部分基于 Windows 风格软件技术如 ActiveX Automation 技术、DDE (动态数据交换)、多线程 (Multithread) 等;
- 3) 功能齐全的软件工具包 (仪器控制、I/O 控制、通讯、数据处理) 等, 有很强的数据处理、数据分析功能;
- 4) 程序自动生成的能力, 减轻软件开发过程中代码编写的工作量;
- 5) 可运行与 Win98/WinNT/Unix/Win2000/WinXP 等系统中。

### 2.2 Lab Windows/CVI 程序的一般结构

Lab Windows/CVI 开发环境构造的一般的测试、控制和信息处理的应用程序具有如图 1 的结构。从

收稿日期: 2003-07-08

作者简介: 刘立月, 男, 安徽望江人, 华东交通大学讲师。

图1中可见,测试应用程序包含以下几部分:

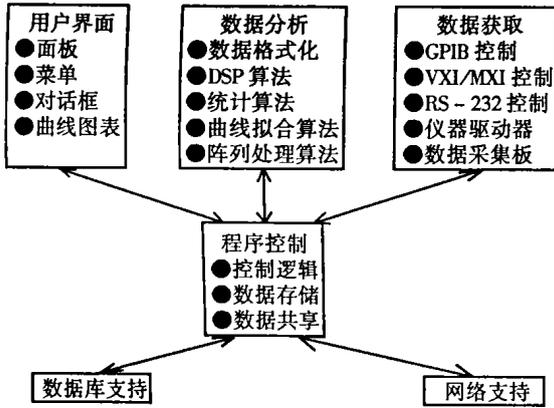


图1 Lab Windows/CVI程序的一般结构

#### 1) 用户界面

通过Lab Windows/CVI提供的用户界面编辑器,程序员可以很方便地通过所见即所得的方式设计并创建自己的操作界面。

#### 2) 控制程序

控制程序是用户程序与用户界面、数据采集、数据分析相关连的部分,包含控制程序流程的控制逻辑和数据存储。Lab Windows/CVI程序中的控制程序部分基本上是由用户自己编写的程序组成。

#### 3) 数据采集

无论数据采集时数据源是什么(外部设备或DAQ板),程序中都需要采集控制逻辑,数据采集为系统提供数据分析部件或其他程序组件所需的原始数据。

#### 4) 数据分析

获得数据后,通常需要对其进行分析。数据分析可分为实时分析和事后分析。数据分析的步骤包含数据格式化、放大或缩小、信号预处理、统计、曲线拟合。数据分析中的各步骤由数据分析系列库函数完成,这些库包括格式化I/O库、分析库、高级分析库。

#### 5) 数据库和网络支持

数据库支持一般是对测试设备、测试信息、测试结果等数据的管理。利用数据库可以更有效地管理各种数据。随着网络技术的发展,分布式数据库在测试系统中也渐渐被应用,使得远程测试成为可能。

其中,前4个部分是一般测试工程所必须的,数据库与网络是一些复杂的测试工程所需要的。各个部分在LabWindows/CVI中都有相应的功能函数库支持。由于LabWindows/CVI在数据库、网络应用开发能力不及其它开发工具(VC、VB、PB等),因此,可

以用其它开发工具(如VC)替代其开发数据库与网络应用程序。

### 2.3 Lab Windows/CVI开发应用程序操作过程

Lab Windows/CVI开发一般测控应用程序过程包括以下几个步骤:

#### 1) 建立工程文件

进入Lab Windows/CVI的开发环境。单击File→New→Project(\*.prj)菜单,新建一个项目工程文件。

#### 2) 建立用户界面文件

单击File→New→User Interface(\*.uir)菜单,新建一个用户界面资源文件,同时弹出用户界面资源的编辑窗口,然后用户可根据应用的需要,进行相应的界面资源的创建与设置。

#### 3) 生成程序代码

Lab Windows/CVI开发环境支持代码生成工具生成程序的主框架,单击用户界面资源的编辑窗口Code→Generate→Main Function……菜单,弹出主函数设置对话框,选择合适的参数,确认后,即可生成所需主函数代码框架。

然后通过函数面板控制项进行消息事件处理,关于Lab Windows/CVI事件驱动处理方法有两种消息事件处理代码框架形式,其相关说明见下一节。

结合应用实际情况,在主函数代码与信息事件处理代码框架中编写程序的逻辑关系与消息处理。

将上述生成的文件加入项目工程文件中。

#### 4) 程序调试和生成可执行文件

Lab Windows/CVI调试工具包括三个方面:代码单独调试;程序断点调试;变量观察。单击项目管理窗口的Build→Target,出现建立目标文件类型的菜单选项,Lab Windows/CVI开发工具支持3种工程:

● EXE 可执行文件工程;

● LIB 静态函数库工程;

● DLL 动态连接库工程。

### 2.4 Lab Windows/CVI的事件驱动处理

Lab Windows/CVI提供两种基本的事件驱动处理方法:回调函数法和事件循环处理法。

回调函数法是开发者为每一个用户界面的控制项写一个独立的函数,当选中某个控制项,就调用相应的函数进行事件处理。一般回调函数的源代码结构如下:

```
int CVICALLBACK AcqData (int panel, int control, int event,
void * callbackData, int eventData1, int eventData2)
```

```

{
switch (event)
{
case EVENT_COMMIT: break;
case EVEN_VAL_CHANGED: break;
case EVENT_LEFT_CLICK: break;
.....
default: break;
}
return 0;
}

```

事件循环处理法使用一个事件循环来处理用户接口的 COMMIT 事件. 在事件循环处理法中只处理用户界面控制项产生的 COMMIT 事件. 通过 GetUserEvent() 函数过滤, 将所有的 COMMIT 事件区分开, 识别是由哪个控制项产生的事件, 并执行相应的处理. 下面是一个简单的事件循环处理法的程序代码结构:

```

While (GetUserEvent (WAIT, &handle, &control ))
{
if ( handle = PANEL)
{
switch ( control )
{
case PANEL_CONTROL1: break;
case PANEL_CONTROL2: break;
.....
}
}
if ( handle = MANUBAR)
{
switch ( control )
{
case MANUBAR_MANU_ITEM1: break;
case MANUBAR_MANU_ITEM2: break;
.....
}
}
.....
}

```

回调函数法和事件循环处理法各有优缺点, 开发驱动程序时应取决于现实应用的情况.

### 3 应用程序实例的 LabWindows/CVI 开发

具体以 PCL-818 卡为例, 利用 LabWindows/CVI 开发平台实现数据实时采集、实时显示、实时存储全过程.

#### 3.1 PCL-818 卡简介

PCL-818 ( PCL-818 High Performance Data Acquisition Card With Programmable Gain ) 是一系列高性能的、高速、可编程、多功能 DAS 卡, 能够提供五种最常用的测量和控制功能: 12 位 A/D 转换、D/A 转换、数字量输入、数字量输出和计数器/定时器.

#### 3.2 应用软件设计

##### 3.2.1 PCL-818 卡面板设计

面板设计是虚拟仪器软件设计出发点, 也是用户操作的图形化接口. PCL-818 数据采集卡仪器面板图设计如图 5 所示, 整个面板由四部分构成: 采集卡配置, 数据处理、数据显示及采集控制.

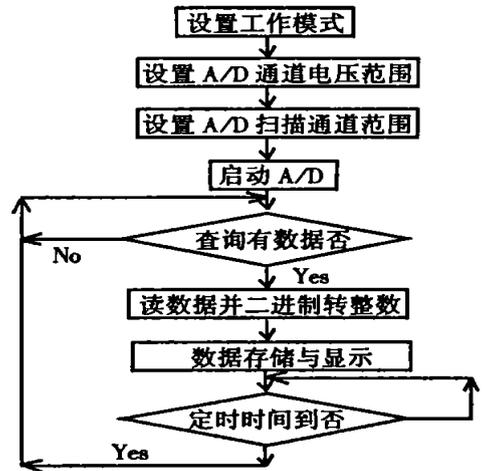


图 3 数据采集流程图

##### 3.2.2 事件驱动程序

LabWindows/CVI 采用 Windows 下编程的事件驱动机制, 它是 PCL-818 卡应用程序设计的关键. 本系统事件有: 定时器溢出; 开始/停止采集键、退出键、数据存储键、数据回放键单击等. 系统在定时溢出事件中执行数据采集程序. 在测试中定时溢出事件同时由开始/停止采集键、退出键等控制. 在编程中, 考虑到事件消息的多样性, 其事件的消息处理采用回调函数法.

##### 3.2.3 数采卡驱动程序

对于 PCL-818 数据采集卡, 由于 Lab Windows/CVI 提供的仪器驱动程序无法直接驱动, 故采用直接操作端口地址法, 用 C 语言提供的端口输入函数 inp()、inpw() 和端口输出函数 outp()、outpw() 对系统板端口直接操作, 以实现底层 I/O 驱动. 数据采集流程图如图 3 所示.

#### 3.3 应用程序运行结果

LabWindows/CVI 开发 PCL-818 卡应用采集程序, 对被测对象实现数据实时采集、实时显示、实时存储全过程. 本实例的被测对象为信号发生器, 信号发生器不断产生正弦波信号, 采集频率为 10ms.

图 4 为应用程序实时采集的数据(自定义文件存储),图 5 为采集数据的实时显示。



图 4 PCL-818 卡实时采集的数据

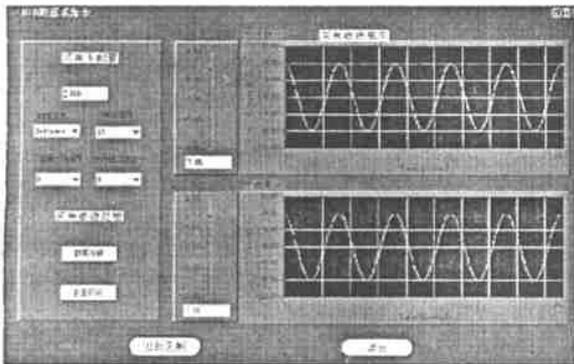


图 5 PCL-818 卡实时显示的数据

## 4 LabWindows/CVI 开发模块与 VC 之间的接口实现

LabWindows/CVI 支持对兼容编译工具开发的外部模块的使用,也能够开发模块供外部编译工具使用。LabWindows/CVI 开发环境支持常见的外部模块接口,包括以下几种:

- 目标代码文件(.obj);
- 静态库(.lib);
- 动态连接库(.DLL);

目标代码文件和静态库中包含了变量、函数的定义和函数的实现,在程序连接时连接到可执行程序;动态连接库则依赖于导出库文件,导出库文

件中包含了函数代码在相应 DLL 文件中的位置。动态连接库的 DLL 文件一般放在操作系统的目录或应用程序目录中,供程序运行时加载。

### 4.1 目标代码文件和静态库的使用

在 LabWindows/CVI 中使用目标代码文件或静态库的方法很简单,直接将兼容的目标代码文件(.obj)或静态库文件(.lib)添加到项目工程中,并在使用函数调用的程序文件中包含相应的头文件即可。工程在连接时,将自动到这些模块中查找相应的函数实现,并连接到可执行文件中。

### 4.2 动态连接库的使用

LabWindows/CVI 开发环境可以使用标准的 32 位动态连接库。它是通过标准的 32 位动态连接库的导出库文件(.lib 文件)连接到 DLL(Dynamic Linked Libraries)文件中的,因此在程序编辑连接的过程中使用的是导出库文件,而不是 DLL 文件。对于 32 位标准动态连接库,其 DLL 文件包含了具体的实现代码,而导出库 lib 文件则包含函数的导出信息,既相应的函数在 DLL 文件中的位置和引用方法等。使用动态连接库的应用程序中没有包含动态连接库中函数的实现代码,在程序连接时只连接了导出库文件中给出的函数位置信息,动态连接库的实现代码是程序运行时从 DLL 文件中加载的。从目前硬件设备驱动程序设计看大都采用 DLL 技术,用 DLL 编写驱动程序,主要基于以下考虑:

- 1) 动态连接库(DLL)实际上是一个函数库。只在应用程序运行期间,DLL 中的函数才被随时调和连接。和静态连接库相比,动态连接库可以和它的应用程序共享库中函数和资源,减少了因重复拷贝而造成的应用程序的冗长以及计算机资源占用;
- 2) 可以保持硬件驱动程序的独立性;
- 3) 容易被其它高级语言编写的应用程序调用,实现应用软件二次开发。

综上所述,可以用 LabWindows/CVI 编制针对硬件数据采集卡的动态连接库模块,并在 VC 环境下调用该动态连接库,从而实现 VC 与采集卡间的通讯。

### 4.3 VC 调用 LabWindows/CVI 开发 DLL 的方法

LabWindows/CVI 开发的包含用户界面文件的 DLL 有一套自己的消息处理方式,为了解决 LabWindows/CVI 的消息与 VC 环境下的消息的冲突问题,在调用 LabWindows/CVI 开发 DLL 时应遵循以下几点:

- 1) LabWindows/CVI 用户界面文件加载、显示及消息处理应作为一个独立的动态连接库输出函数

处理,这样可以确保 LabWindows/CVI 界面消息的独立性,其设计思想类似于 VC 环境下对外部进程或线程的调用.以下为 818 卡动态连接库调用的函数:

```
void DLLEXPORT RUN818DLL (void)
{
    .....
    mainPanel=LoadPanelEx(0, "818. uir", MAIN _ PANEL, _
CVIUserHInst);
    DisplayPanel (mainPanel);
    RunUserInterface ();
}
```

2) 在 LabWindows/CVI 用户界面退出的处理时,应保证其用户界面的隐藏、删除及结束消息的处理.以下为 818 卡用户界面退出的处理函数:

```
int CVICALLBACK AcqExit (int panel, int control, int event,
void * callbackData, int eventData1, int eventData2)
{
    .....
    HidePanel(mainPanel);
    DiscardPanel(mainPanel);
    QuitUserInterface (0);
    .....
}
```

3) 在 LabWindows/CVI 动态连接库生成中包含 \*.dll 与 \*.lib 文件, \*.dll 文件包含了具的实现代码,而导出库 \*.lib 文件则包含函数的导出信息.不同的开发环境性 \*.dll 大都相同,但导出库 \*.lib 文件可能不一致,因此在 LabWindows/CVI 中,必须利用其工具导出与 VC 环境下兼容的导出库 \*.lib 文件.这样其开发的动态连接库方可被 VC 调用.

## 5 LabWindows/CVI 应用程序发布

LabWindows/CVI 开发环境生成的应用程序或

动态连接库(.DLL),必须在 CVI 引擎(CVI 引擎实际上就是一些运行 LabWindows/CVI 应用程序或调用 LabWindows/CVI 开发的 DLL 所必须的驱动 DLL)的驱动下才能运行或使用.如果计算机上已经安装过 LabWindows/CVI,则自动安装了 CVI 引擎,LabWindows/CVI 开发工具开发的应用程序可以单独运行;如果只是将 LabWindows/CVI 开发的应用程序及相关文件(用户界面资源、用户库文件等)直接拷贝到没有安装 CVI 引擎的计算机上,则无法运行.

在 LabWindows/CVI 开发环境中,提供了 LabWindows/CVI 的安装程序制作工具,用户可利用此工具对应用程序或动态连接库进行打包.在没有安装 CVI 引擎的其它计算机上直接运行已打包的程序.

## 6 结束语

文章首先分析 LabWindows/CVI 的特点及其开发一般测控系统软件的过程;然后具体以 PCL-818 卡为例,结合其特点,在 LabWindows/CVI 中,利用直接操作端口地址法开发并实现了数据实时采集、实时显示、实时存储全过程.最后就有关 LabWindows/CVI 开发模块与 VC 之间的接口问题及 LabWindows/CVI 的应用程序发布进行了阐述.

### 参考文献:

- [1] "LabWindows/CVI User Interface Reference" National Instruments Corporation. 1998.
- [2] "LabWindows/CVI User Programmers Reference" National Instruments Corporation. 1999.
- [3] "LabWindows/CVI User Instrument Driver Developer's Guide" National Instruments Corporation. 1999.
- [4] 张风均. LabWindows/CVI 开发入门和进阶[M]. 北京:北京航空航天大学出版社,2001.

(下转第 96 页)

的 2 输入端.

(Connected (Out 1 A1) (In 1 R1)): A<sup>1</sup> 的输出端连入 X<sup>2</sup>

的 1 输入端.

其余的连结是:

(Connected (Out 1 X<sup>2</sup>) (Out 1 C1))

(Connected (Out 1 R1) (Out 2 C1))

(Connected (In 1 C1) (In 1 X<sup>1</sup>))

(Connected (In 1 C1) (In 1 A<sup>1</sup>))

(Connected (In 2 C1) (In 2 X<sup>1</sup>))

(Connected (In 2 C1) (In 2 A<sup>1</sup>))

(Connected (In 3 C1) (In 2 X<sup>2</sup>))

(Connected (In 3 C1) (In 1 A<sup>2</sup>))

根据上面的表示过程, 我们知道使用 KIF 进行知识表示是一个分层定义和逐步包装的过程.

## 5 结 束

这里我们实现了 1bit 全加器的 KIF 知识表示过

程, 当然我们可以将任何复杂问题, 如较为复杂的数字电路使用 KIF 表示成机器可以理解的知识, 不仅可以实现计算机对数字电路逻辑推理, 更为重要的是 KIF 格式保证了不同的计算机系统之间进行知识交换, 从而实现不同机器系统之间都可以实现知识的共享和处理. 有些推理机都实现了 OKBC 接口功能, 利用 KIF 表示的知识进行推理, 从而实现问题的计算机化处理.

### 参考文献:

- [1] 林尧瑞, 等. 专家系统与原理[M]. 北京: 清华大学出版社, 1988.
- [2] KIF Knowledge Interchange Format, <http://www.csee.umbc.edu/kif/>.
- [3] 钱培怡, 等. 数字逻辑电路的描述及模块化综合方法[J]. 系统工程与电子技术, 2002, (24): 3.

## Knowledge Representation about Digital Circuit with KIF

LIU Zun-xiong, XU Xiang-zheng, NIE Guo-xing

(School of Electric and Electronic Eng., East China Jiaotong Uni., Nanchang 330013, China)

**Abstract:** KIF (Knowledge Interchange Format) is an accepted common knowledge representation language. The paper introduces KIF and its role, and gives the steps about how to use KIF with a simple digital circuit.

**Key words:** digital circuit; KIF; machine understanding

(上接第 84 页)

## Application of Lab Windows/CVI in Measurement and Control System and Interface Implementation

LIU Li-yue<sup>1</sup>, TANG Wen-liang<sup>1</sup>, FU Jun-dong<sup>2</sup>

(1. School of Information Engineering, East China Jiaotong University, Nanchang, Jiangxi, 330013; 2. School of Electrical and Electronic Engineering, East China Jiaotong University Nanchang Jiangxi, 330013, China)

**Abstract:** The paper, taking an example of software development of PCL-818 card, expatiates on the whole course of measurement & control's software in Experiment & Measurement System developed by Lab Windows/CVI. In view of conflict problem in message mechanism between Lab Windows/CVI development kits and Visual C++ Development Environments, the paper suggests that a design thinking of process encapsulation can be used to solve message conflict and Lab Windows/CVI development module is used effectually in Visual C++ development environments.

**Key words:** Lab Windows/CVI; Visual C++ development environments; experiment & measurement system